Optimization-Based Algorithms for a Single Level Constrained Resource Problem

So Wai Kuen

The Chinese University of Hong Kong

July 1996

Abstract

In this research, we formulate and solve a class of Single Level Constrained Resource (SCLR) problem. The objective is to develop an aggregate production plan to minimize the sum of setup costs, regular time production costs, overtime production costs, and inventory holding costs subject to resource constraints. The problem has been shown to be NP-hard in computational complexity.

We first consider the SLCR problem with backordering. Many existing formulations for the problem only consider a single source of production capacity. Our formulation is more comprehensive because it allows for multiple sources of production capacity. To solve the problem, we develop a heuristic based on the special structure of fixed charge transportation problem. We establish the performance of our algorithm by finding Lower Bound value and by comparing to the algorithm by Millar and Yang (1994).

We then restrict our formulation to disallow backordering. The problem without backordering was studied extensively in Gilbert and Madan (1991). Based on the idea by Gilbert and Madan, we develop an algorithm that outperforms the one by Gilbert and Madan.

Acknowledgment

I would like to thank all of people who have helped and guided me through my research work.

I would like to express my deep gratitude and appreciation to Dr. C.H. Cheng, my supervisor, for his expert help, encouragement, and patience in the completion of this research. Also, I would like to thank Dr. Manu S. Madan for his assistance and guidance.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This research deals with Single Level Constrained Resource (SLCR) problem. The SLCR problem (with and without backordering) is formulated as a variation of the fixed charge transportation problem. We develop heuristic algorithms for both cases. In addition, we present results of extensive computational tests.

1.1 Introduction to SLCR Problem

The SLCR problem is concerned with the planning of production for multiple products in multiple period to satisfy demands without violating resource constraints. The problem can be found in many manufacturing settings. One such example involves developing a master production schedule at a plant producing rolled aluminum coils. The SLCR problem is shown to be NP-hard by Gilbert and Madan (1991). Most solution algorithms rely on heuristic methods. These heuristic algorithms will be reviewed in Chapter 2 of this thesis.

1.2 Our Contributions

We first consider the formulation of the SLCR problem to allow for backordering. Very few researchers have considered this problem in association with backordering. For those who have taken backordering into consideration, they only included an arbitrary set of manufacturing factors in their formulation. Typically most formanulations in the literature are restrictive in the sense that they allow for only one type of production capacity i.e. regular time. Our formulation is more comprehensive in nature as it allows for different types of production capacities such as regular time, overtime and subcontracting.

The formulated problem carries a structure of fixed charge transportation problem. Based on the special problem structure, we develop an efficient heuristic algorithm. The application of fixed charge transportation problem to the SLCR problem was presented by Gilbert an Madan (1991). However, approach did not address the backordering issue. Out computational results show that our algorithm produces very reliable results. Also, we establish the performance of our algorithm by comparing to the algorithm by Millar and Yang (1994).

We will then restrict our formulation to disallow backordering. Based on the fixed charge transportation problem structure, we develop an efficient heuristic. Our computational results show that our algorithm outperforms the one developed by Gilbert and Madan (1991).

1.3 Organization of the thesis

This chapter provides a brief introduction to our research problem. Chapter 2 presents a literature review of solution algorithms for the problem. In Chapter 3, we formulate the SLCR problem with backordering as the fixed charge transportation problem. A heuristic algorithm is developed using the fixed charge transportation problem structure. A Lower Bound procedure is also developed to evaluate the quality of solution generated by our algorithm. Besides, our algorithm is compared to the one developed by Millar and Yang. In Chapter 4, we propose a heuristic algorithm for the SLCR problem without backordering. Then we design an experiment to evaluate our algorithm. The algorithm by Gilbert and Madan (1991) is also implemented for comparison purpose.

Chapter 2

Literature Review

This research deals with a master production planning problem. In this chapter, we review the literature relevant to this research. First, we describe work related to capacitated resource constraint problem. In this research, we use fixed charge transportation problem in the design of our heuristic algorithms. Therefore, we also discuss the fixed charge problem.

2.1 Research in the Capacitated Resource Constraint Problem

In the literature, there are two approaches to deal with the problem: Single Level Constrained Resource (SLCR) problem and Multiple Level Constrained Resource (MLCR) problem. Both types of the problem have extensively studied. We only present some representative work done in the area.

2.2 The Single Level Constrained Resource Problem

Most researchers in the area use linear programming based on dynamic programming based algorithms to solve their formulated problems.

Manne (1958) proposes an algorithm based on set partitioning. He relaxes a 0-1 set partitioning problem to a linear programming problem and rounds off the resulting solutions. The advantage of his approach is that the formulation can be solved with linear programming.

Dzielinski and Gomory (1965) apply Dantzig-Wolfe (1961) decomposition principle to improve the efficiency of the Manne's approach. Lasdon and Terjung (1967) propose their column generation technique and a generalized upper bounding procedure by Dantzig and Van Slyke (1967) to replace Dantzig-Wolfe decomposition principle. Their analysis shows that the combined techniques are more efficient than Dantzig-Wolfe decomposition principle.

Bahl (1983) proposes two algorithms. Both algorithms use column generation techniques by Lasdon and Terjung (1967). Instead of using Wagner-Whitin (1958) condition in column generation, he uses a simple lot sizing procedure.

Barany et al. (1984) develop the solution algorithm exploiting concept of cutting planes and branch and bound. Eppen and Martin (1987) also propose the algorithm using cutting planes and branch and bound.

Pochet and Wolesey (1988) propose two strong reformulations of the problem. The first one is a shortest-path reformatulation, very similar to the one proposed by Epper and Martin (1987). The formulation is solved by a standard integer programming package. The second reformulation uses a plant location formulation. The formulated problem is solved by a cutting plane algorithm.

More recently, Gilbert and Madan (1991) formulate the SLCR problem as fixed charge transportation problem. Different from most formulations, their formulation considers both the regular time and overtime capacity. The problem is shown to be NP-hard. They develop an optimization-based algorithm for their problem which exploits the special problem structure. Gilbert and Madan (1992) discuss an exact solution algorithm for the SLCR problem without backordering.

Miller and Yang (1994) also consider the problem with backordering. They develop a solution algorithm using Lagrangain decomposition and Lagrangain relaxation techniques. However, their formulation does not take overtime capacity into consideration.

Hindi (1995) develops a model based on variable redefinition and uses a branch and bound algorithm to solve the model. The efficiency of his algorithm is achieved by sharp lower and upper bounds.

Some researchers use the dynamic approach. Zangwill (1966) formulates the problem as a network problem and uses the concave cost network analysis.

Silver and Meal (1973) solve a problem with capacity constraints for a one product case. Their algorithm combines production for several periods into one lot if average cost per period for those periods is decreasing.

Eisenhut (1975) relies on marginal analysis. Holding cost and setup cost are considered. His dynamic algorithm determines lot sizes for each period. The advantage of his algorithm is that it allows for uncertain and fluctuating demand. The problem with his algorithm is that it does not always give a feasible solution.

Lambrecht and Vanderveken (1979) try to correct the infeasible problem of the Eisenhut algorithm. They improve marginal cost determination and incorporate feasibility assurance procedure. Dixon and Silver (1981) develop a solution algorithm using a "Greedy" approach. Their algorithm always guarantees a feasible solution.

Dogramaci et al. (1981) develop an algorithm to find a schedule for a product. Their algorithm includes some feasibility seeking steps and solution improvement steps. Karni and Roll (1982) rely on the Wagner-Whitin (1958) solution for a product and find a better feasible solution by some heuristic methods.

2.3 The Multiple Level Constrained Resource Problem

Lambrecht and Vanderveken (1978b) develop two heuristic algorithms for the multiple level constrained resource problem. The capacity constraint is only enforced at the highest level.

Billington et al. (1981) propose a mixed integer linear programming formulation considering setup cost, and overtime capacity constraints. In their follow-up work, Billington et al. (1983) propose another formulation that allows a setup for each product. In their final work, Billington et al. (1984) develop a branch and bound procedure using the idea of Lagrangain relaxation.

Zahorik et al. (1984) assume that capacity is constrained only at one level. They develop an optimizing algorithm for the 3-period problem and a heuristic algorithm for the problem with more than 3 periods.

2.4 Research in the Fixed Charge Problem

The fixed charge problem assumes that a fixed cost is incurred whenever the associated continuous variable becomes positive. In the fixed charge transportation problem, each cell in the transportation tableau is assumed to have fixed cost and variable cost. The SLCR problem proposed by Gilbert and Madan (1991) and considered in this thesis is formulated as a variation of the fixed charge transportation problem. A fixed charge is shared by a group of cells in the transportation tableau. This requirement makes many existing algorithms for the fixed charge transportation problem impractical. However, we will still review some representative work here.

2.4.1 Approximate Methods

Hirsch and Dantzig (1968) show that for any fixed charge problem, an optimal solution can be found in an extreme point of the constraint set. Balinski (1961) replaces the non-linear fixed charge objective function with an approximate linear objective function in solving the fixed charge transportation problem. Then he solves the modified problem by a transportation algorithm.

Cooper and Drebes (1967) propose an algorithm employing an extreme point heuristic. The algorithm improves a solution by removing the variables with largest fixed charge from the basis and inserting the variables with the smallest fixed charge.

Denzler (1969) provides an algorithm using the simplex method. At each iteration, a fixed charge must be considered when determining variables entering the basis or exiting the basis.

In order to improve the computational efficiency, Steinberg (1970) develops an adjacent extreme point heuristic based algorithms. The algorithm attempts to find better solutions by moving away from the current local minimum in as few iterations as possible.

Walker (1976) extends the Denzler's algorithm to consider degeneracy. His heuristic algorithm first finds an initial local minimum and searches for an improved solution by considering adjacent extreme points.

Disby (1991) proposes a successive linear approximation procedure for the fixed charge transportation problem. Wright etal. (1991) propose a new heuristic to efficiently solving the fixed charge problem.

2.4.2 Exact Methods

Although exact algorithms are not effective for solving the large fixed charge problem, they are useful for evaluating approximate algorithms. Murty (1968) solves the linear fixed charge problem by ranking the vertices of the polyhedral constraint set in increasing order of their continuous objective values. Then he adds fixed charges to determine an optimal solution. Gary (1968) decomposes the fixed charge problem into a master integer problem and a series of transportation subproblems.

Kennington (1976) proposes a branch and bound algorithm and develops some good bounds.

Kennington and Unger (1976) transform the fixed charge transportation problem into another equivalent problem using a network structure. They develop a branch and bound algorithm for solving the equivalent problem. McKeown (1981) develops a branch and bound algorithm. Bounds are obtained by calculating separately on the sum of fixed costs and on the continuous costs. Cabot and Erenguc (1984) present a branch and bound algorithm using Langragian relaxation.

McKeown and Ragsdale (1990) propose a stronger formulation of the fixed charge problem. They also include a precise step to improve the computational efficiency. Palekar etal. (1990) develop a new branch and bound algorithm for the fixed charge transportation algorithm. The computational efficiency is achieved using good bounds.

2.5 Conclusion

In this research, we will base on the Gilbert and Madan (1991) formulation of the SLCR problem. The underlying fixed charge transportation problem structure assumes that a group of cells in the transportation tableau share a fixed charge. This is different from the traditional fixed charge transportation problem where a fixed charge is assigned to each cell in the transportation tableau. Therefore, existing algorithms for the fixed charge transportation problem cannot be directly applied to solve the SLCR problem.

Chapter 3

The SLCR Problem with backordering

As discussed in Chapter 2, the SLCR problem has evoked considerable interests among the researchers. Many heuristic algorithms have been developed for this problem and its variations. Although backorder situations commonly arise in manufacturing settings, few researchers consider the traditional SLCR problem with backorder cost (Zangwill, Pochet and Wolsey 1988, Millar and Yang 1994). These research studies were already reviewed in Chapter 2.

A recent study by Millar and Yang (1994) develops a heuristic based on Lagrangain decomposition and relaxation for the SLCR problem with backordering. Their model takes into consideration holding cost, setup cost, and backorder cost. Overtime production cost is not considered in the model. In this chapter, we develop a comprehensive model which considers regular production cost, and overtime cost, in addition to holding cost, setup cost, and backorder cost. Our model allows for overtime production, as overtime production is commonly used as a means for temporarily adjusting capacity.

A basis for our research is that the SLCR problem with backordering and overtime production can be formulated as a variation of fixed charge transportation problem. The heuristic, exploiting the underlying special structure of the problem, uses the following facts:

- Each cell in the transportation tableau is assumed to have fixed cost and variable costs.
- A fixed charge is shared by a group of cells in the transportation tableau.

In the primal simplex algorithm for the transportation problem, the pivot rule chooses the non-basic variable to enter the basis to bring the largest decrease in the objective function value. The algorithm terminates if there is no more improvement by bringing a non-basic variable into the basis.

3.1 Problem Description and Formulation

The SLCR problem considers J different products and T equal periods. The types of production capacity available are regular time and overtime production. The number of units of capacity used in the production of a given product is directly proportional to the quantity produced. Thus for purposes of exposition, all quantities are expressed in units of capacity required. For product j in period m, $j = 1,2,\ldots,J$, $m = 1,2,\ldots,T$, the demand, the production level, and the end of period inventory level, will be expressed in units of capacity, i.e., one unit of a product is the amount that can be produced using one unit of capacity.

Let k denote types of capacities, $k = 1,2,...,K$. There are two capacity source, regular time (k=1) and overtime (k=2). Each product j, $j = 1,2,..,J$, has a holding cost of H_j per unit per period. Similarly, each product j, $j = 1,2,...,J$, has a backorder cost of B_j per unit per period. Each product requires a setup for each period in which it is produced. The cost of a setup for product j is S_j units per setup, $j = 1,2,...,J$. It is assumed that downtime consumed by the setup operation is negligible.

Let j index product, k index source of production capacity, m index period in which product is produced and n index period in which product is demanded, $j = 1, 2, ..., J$, $k = 1, 2, ..., K$, $m = 1, 2, ..., T$, $n = 1, 2, ..., T$.

Let,

J = number of products;

K = number of type of production capacity available;

T = number of time periods in the planing horizon;

D_{jn} = demand of product j in period n;

C_{km} = units of capacity type k available in period m;

S_j = setup cost for product j;

H_j = holding cost in dollars per period of product j;

B_j = backorder cost in dollars per period of product j;

P_k = production cost per unit of capacity type k;

Y_{jm} = a binary variable where indicate the presence or absence if setup for product j in period m.

Let X_{jkmn} be the quantity, expressed in units of capacity, of product j produced using capacity source k in period m used to meet the demand in period n, where, j = 1,2,...,J, k = 1,2,...,K, m = 1,2,...,T, and n = 1,2, ...,T. The formulation of the SLCR problem with backorders is, given as follows,

Minimize :

$$\sum_{j=1}^{J}\sum_{m=1}^{T} S_j * Y_{jm} + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{m=1}^{T}\sum_{n=1}^{T} X_{jkmn} * P_k + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{m=1}^{T}\sum_{n=m}^{T} (n-m) * X_{jkmn} * H_j$$
$$+ \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{m=1}^{T}\sum_{n=1}^{m} (m-n) * X_{jkmn} * B_j \qquad (3.1)$$

Subject to :

$$\sum_{k=1}^{K}\sum_{m=1}^{T} X_{jkmn} = D_{jn} \qquad j = 1,2,...,J \quad n = 1,2,...,T \qquad (3.2)$$

$$\sum_{j=1}^{J}\sum_{n=1}^{T} X_{jkmn} \leq C_{km} \qquad k = 1,2,...,K \quad m = 1,2,...,T \qquad (3.3)$$

$$Y_{jm} = \begin{cases} 1 \text{ if} \sum_{k=1}^{K}\sum_{n=1}^{T} X_{jkmn} > 0 \\ 0 \text{ otherwise} \end{cases} \qquad j = 1,2,...,J \quad m = 1,2,...,T \qquad (3.4)$$

$$X_{jkmn} \geq 0 \qquad j = 1,2,...,J \quad k = 1,2,...,K \quad m = 1,2,....,T \quad n = 1,2,...,T \qquad (3.5)$$

Problem (3.1) to (3.5) is a variation of the fixed charge transportation problem. A group of cells in the transportation tableau shares a fixed charge. This feature makes our formulation different from the traditional fixed charge transportation problem.

Like a transportation problem, (3.1) to (3.5) can be represented by a transportation tableau, in which rows are supplies (production capacities) and columns are demands. It turns out that the transportation tableau is very helpful in designing our solution algorithm. In addition, we feel that the transportation tableau is also a good decision tool. With the transportation tableau, we can easily explain a production schedule to plant managers and workers.

The first term in the objective function (3.1) represents the sum of setup costs for all products in every period, the second term represents total production capacity cost, and the third term represents total inventory holding cost. The fourth term here, represents the total backorder cost.

Constraints (3.2) are demand constraints which require that demand, D_{jn}, of product j, $j = 1,2,...,J$, during period n, $n = 1,2,...,T$, is met by X_{jkmn}. The sum of production quantity of product j in period m (m is from 1 to T) is used to meet the demand of period n.

Constraints (3.3) are capacity constraints which require that $\sum_{j=1}^{J}\sum_{n=1}^{T} X_{jkmn}$, the total amount of production capacity of type k used in period n (n is from 1 to T) to produce different products, does not exceed the total amount of capacity of type k available in period m, $m = 1,2,...,T$; $k = 1,2,...,K$.

Usually the total demand and the total supply are equal. The problem is infeasible if the total demand is greater than the total supply. However, in cases without backordering, we have additional information, i.e. the problem is infeasible if total demand for a given period is greater than total available capacity of the given period. If the total demand is less than the supply, then an additional column (dummy column) with demand equaling excess supply is added.

If total demand for products over the planning periods is less than the total available capacity, we define $D_{J+1\ T+1}$ as follows:

$$D_{J+1T+1} = \sum_{k=1}^{K}\sum_{m=1}^{T} C_{km} - \sum_{j=1}^{J}\sum_{n=1}^{T} D_{jn}$$

where $D_{J+1\ T+1}$ is demand for the dummy column ($J+1$, $T+1$) (destination). We then add the following demand constraint for dummy demand.

$$\sum_{k=1}^{K}\sum_{m=1}^{T} X_{J+1kmT+1} = D_{J+1T+1}$$

Constraint (3.3) will be changed to equality constraints as shown below:

$$\sum_{j=1}^{J}\sum_{n=1}^{T} X_{jkmn} + X_{J+1kmT+1} = C_{km} \qquad k = 1,2,...,K \quad m = 1,2,...,T$$

Constraints (3.4) are logical constraints which account for setups for product j in period m. A setup cost S_j is incurred once for product j in period m if the product is produced in period m (using any type of capacity). In other words, if $\sum_{k=1}^{K}\sum_{n=1}^{T} X_{jkmn}$ is positive, $j = 1,2,...,J$; $m = 1,2,...,T$, then Y_{jm} is One.

Constraints (3.5) are non-negativity constraints.

Figure 3.1 is a tranportation tableau associated with the formulated problem. A supply in the tableau is denoted by C_{km} which is the quantity if source k production capacity available in period m. A demand is denoted by D_{jn} which is the quantity if product j demanded in period n.

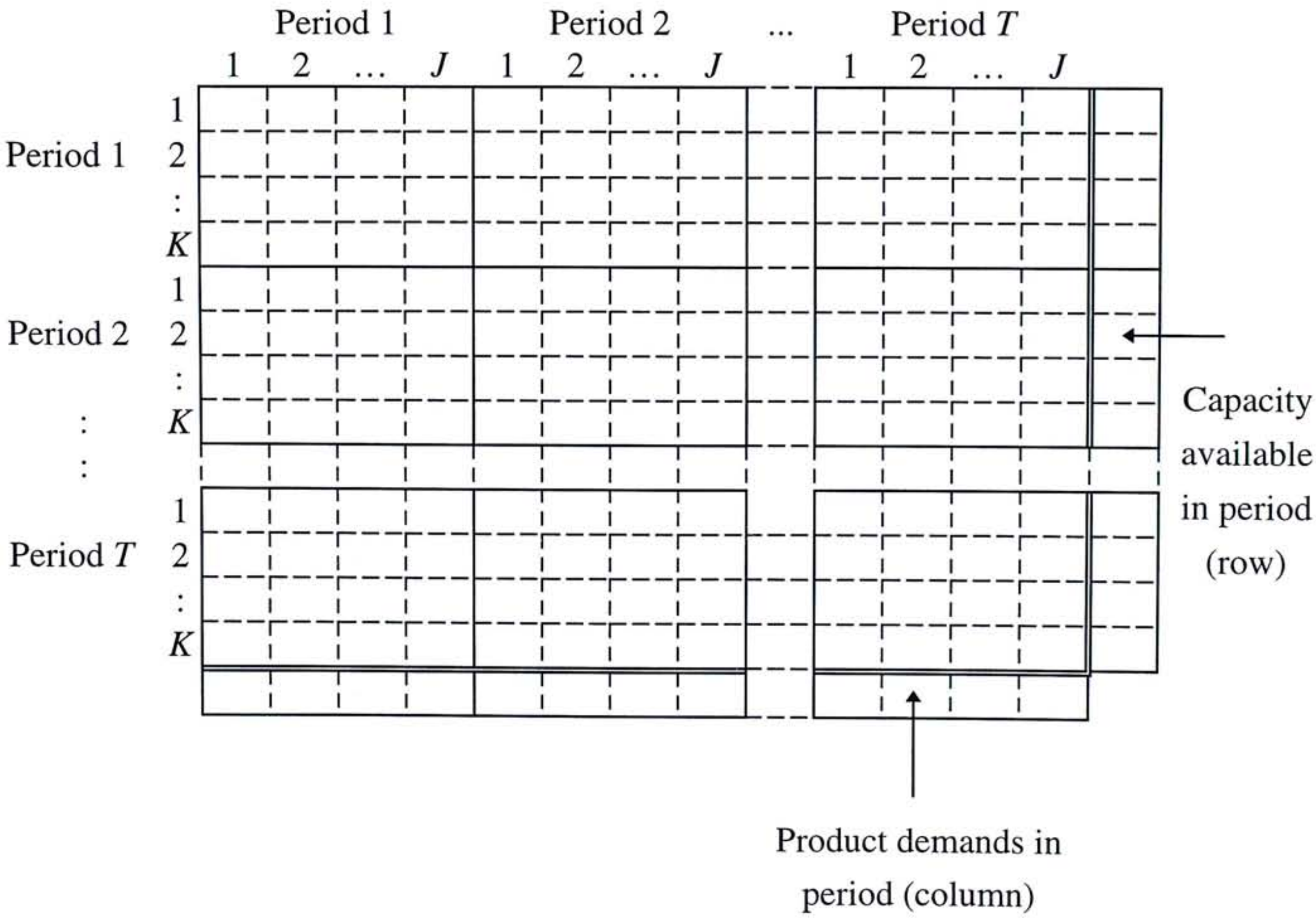


Figure 3.1

An example of transportation tableau with backordering

The cell in row (k, m) and column (j, n) is denoted as cell ((k, m), (j, n)). The flow in this cell corresponds to the variable X_{jkmn}. The value of the setup variable Y_{jm} is implicitly determined by constraint (3.4). Hence a feasible tableau solution corresponds to a feasible solution to problem (3.1) to (3.5).

3.2 Description of the heuristic

In this section, the heuristic for solving the SLCR problem with backordering will be presented. The heuristic works with basic feasible solutions to the transportation problem (3.1) to (3.5). The heuristic consists of two phases. The first phase is a single pass "Greedy" algorithm that selects the basic variables to provide a good starting solution. The second phase makes an attempt to improve the solution obtained in the first phase by replacing variables in the basis.

As backordering is allowed, the determination of setups necessary to satisfy demands in the first period is not straight forward.

3.2.1 Phase I

This new heuristic is based on the Vogel's Approximation Method (Reinfeld and Vogel 1958) and is designed to take into account fixed cost associated with each cell in the tableau.

According to the Vogel's Approximation Method, penalty is calculated to indicate where departure from lowest cost allocations will bring the highest increase in cost. Instead of finding the penalty using the real production costs, we use the nominal cost (which takes into account the fixed cost) to calculate the penalty. The maximum possible value is assigned to the variable associated with the cell having the largest penalty.

For any cell ((k,m) (j,n)) which is chosen to enter the basis, X_{jkmn} will take on a value equal to the minimum of $\overline{D_{jn}}$ or $\overline{C_{km}}$.

The demand of period n can be met by producing the products in previous periods ($m < n$), or by producing the products in the same period, or by producing products in the later periods ($m > n$). There will be an increase of holding cost per unit or backorder cost per unit by $(n\text{-}m) * H_j$ or $(m\text{-}n) * B_j$, respectively. The nominal cost R_{jkmn} is calculated by using the original idea from the Madan and Gilbert heuristic,

$$P_k + (n-m) * H_j + \frac{S_j}{Min(\overline{D_{jn}}, \overline{C_{km}})}$$

We introduce a new term θ,

$$\text{where } \theta = \frac{S_j}{D_{jn} * Min(\frac{\sum_{t=1}^{n-1} H_j * t}{n-1}, \frac{\sum_{t=n+1}^{T} B_j * (t-n)}{T-n+1})}$$

and $\sum_{t=1}^{n-1} H_j * t$ is the total holding cost to produce products in period 1 to n-1 to meet the demand in the coming period n, $\sum_{t=n+1}^{T} B_j * (t-n)$ is the total backorder cost to produce products in period n+1 to T to meet the demand of the previous period n. Since we do not know how many periods should be considered to hold the products or how many periods after the current period should be used to produce the products, we consider the average holding costs and backorder costs. θ is used to reflect whether it is worth scheduling the products to be produced in the current period n incurring setup cost or avoiding setup by producing the products in previous periods or later periods. It is clear that if the setup cost is high, products should be scheduled for previous or later time periods. Hence, a setup is scheduled for product j in period n, if $\theta < 1$. The detailed steps in Phase I are given in the following.

Step 1: *Computation of nominal cost for cells in tableau*

Compute the nominal cost R_{jkmn} for admissiable cells $((k,m)\ (j,n))$,

For $n > m$,

$$R_{jkmn} = P_k + (n - m) * H_j \qquad j = 1,2,\ldots,J$$

$$\text{if } \sum_{q=1}^{K}\sum_{t=1}^{T} X_{jqmt} > 0 \qquad k = 1,2,\ldots,K$$

$$R_{jkmn} = P_k + (n-m) * H_j + \frac{S_j}{Min(\overline{D_{jn}}, \overline{C_{km}})} \qquad m = 1,2,\ldots,T \qquad n = 1,2,\ldots,T$$

$$\text{if } Min(\overline{D_{jn}}, \overline{C_{km}}) > 0 \text{ AND } \sum_{q=1}^{K}\sum_{t=1}^{T} X_{jqmt} = 0$$

For $m > n$,

$$R_{jkmn} = P_k + (m - n) * B_j \qquad j = 1,2,\ldots,J$$

$$\text{if } \sum_{q=1}^{K}\sum_{t=1}^{T} X_{jqmt} > 0 \qquad k = 1,2,\ldots,K$$

$$R_{jkmn} = P_k + (m-n) * B_j + \frac{S_j}{Min(\overline{D_{jn}}, \overline{C_{km}})} \qquad m = 1,2,\ldots,T \qquad n = 1,2,\ldots,T$$

$$\text{if } Min(\overline{D_{jn}}, \overline{C_{km}}) > 0 \text{ AND } \sum_{q=1}^{K}\sum_{t=1}^{T} X_{jqmt} = 0$$

Step 2: *Introduction of setup for those period with low setup cost*

For each unmarked column (*j*,*n*), using source *k*,

$$\text{compute } \theta = \frac{S_j}{D_{jn} * Min\left(\frac{\sum_{t=1}^{n-1} H_j * t}{n-1}, \frac{\sum_{t=n+1}^{T} B_j * (t-n)}{T-n+1}\right)}$$

$R_{jknn} = 0$ if $\theta < 1$ (It is highly recommended to introduce a setup in the current period)

Step 3: *Computation of row penalty*

For each unmarked row compute penalty E_{km}

where E_{km} is the value difference between the lowest nominal cost and the second lowest nominal cost in the row (*k*,*m*)

Step 4: *Computation of column penalty*

For each unmarked column compute penalty G_{jn}

where G_{jn} is the value difference between the lowest nominal cost and the second lowest nominal cost in the column (*j*,*n*)

Step 5: *Selection of cell to enter basis*

Let $E_{k'm'} = Max(E_{km})$

and $G_{j'n'} = Max(G_{jn})$

IF $E_{k'm'} > G_{j'n'}$

$R_{j'k'm'n'} = Min(R_{jk'm'n})$ $\quad j = 1,2,...,J\,;\, n = 1,2,...,T$

ELSE

$R_{j'k'm'n'} = Min(R_{j'kmn'})$ $\quad k = 1,2,...,K\,;\, m = 1,2,...,T$

where $((k',m')\ (j',n'))$ is the cell selected to enter the basis.

Step 6: *Assign value to cell* $((k',m')\ (j',n'))$

$X_{j'k'm'n'} = Minimum(\overline{D_{j'n'}}\ ,\ \overline{C_{k'm'}}\ )$

IF $X_{j'k'm'n'} = 0$

Mark column (j',n') and loop back to Step 1

ELSE

Proceed to Step 7

Step 7: *Update row capacity and column demand*

IF $\overline{D_{j'n'}} \le \overline{C_{k'm'}}$

$$\overline{C_{k'm'}} = \overline{C_{k'm'}} - \overline{D_{j'n'}}$$

$$\overline{D_{j'n'}} = 0$$

Mark Column (j',n') (To indicate that the corresponding demand has been met)

ELSE

$$\overline{D_{j'n'}} = \overline{D_{j'n'}} - \overline{C_{k'm'}}$$

$$\overline{C_{k'm'}} = 0$$

Mark Row (k',m') (To indicate that the corresponding capacity has been completely used)

Step 8: *Re-compute the nominal costs*

IF any column or row are unmarked (means demands are not met)

Re-compute the nominal costs for cells which share a fixed charge with the cell $((k',m')\ (j',n'))$.

$R_{j'km'n} = P_k + (n\text{-}m') * H_{j'}$ $\quad k = 1,2,\ldots,K\ ;\ n = m', m'+1,\ldots,T$

and $R_{j'km'n} = P_k + (m'\text{-}n) * B_{j'}$ $\quad k = 1,2,\ldots,K\ ;\ n = 1,2,\ldots,m'\text{-}1$

Loop back to Step 1

ELSE

Proceed to Phase II

3.2.2 Phase II

The heuristic used in this phase is similar to the improvement step of Gilbert and Madan's algorithm. The aim of this phase is to reduce the setup cost and the production cost to obtain the lower objective function value.

We introduce non-basic cells into basis at each iteration, to reduce the total cost by ΔZ, where ΔZ is the sum of ΔS and ΔV,

$$\Delta Z = \Delta S + \Delta V$$

where ΔS is the setup cost difference when a new non-basic cell enters basis;

ΔV is the variable cost difference when a new non-basic cell enters basis.

In order to final a better solution, we search for the non-basic cells from four directions (Figure 3.2) and choose the best solution.

(1) Forward across cells - increase column and then increase row;

(2) Forward down the cells - increase row first and then increase column;

(3) Backward across cells - decrease column and then decrease row;

(4) Backward up cells - decrease row and then decrease column.

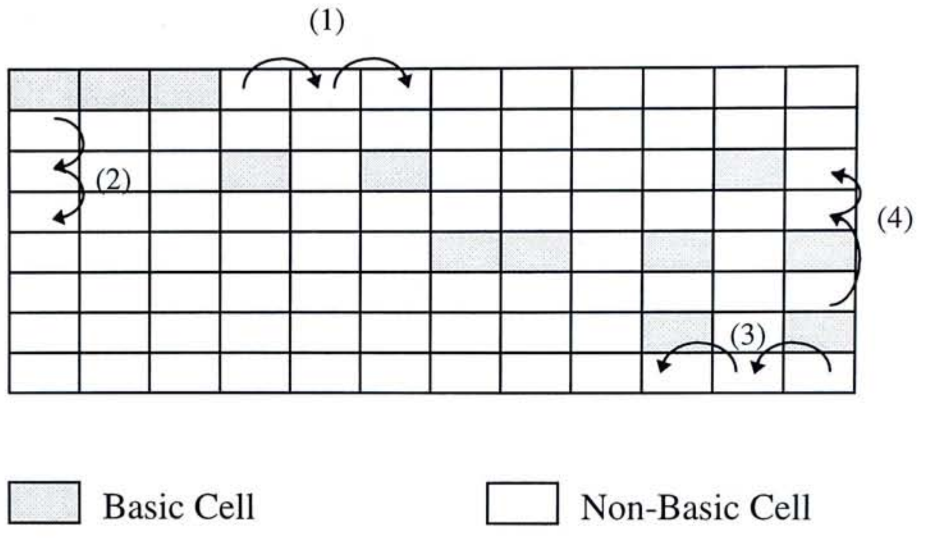


Figure 3.2

Four directions search

The detailed steps in Phase II are shown in the following.

Step 1: *Initialize the searching direction i (i = 1,2,3,4)*

1 = Forward Across,

2 = Forward Down,

3 = Backward Across and

4 = Backward Up

Initialize $i = 1$

Step 2: *Search for non-basic cells*

Find a non-basic cell in the tableau by following the direction i

Step 3: *Calculation of the difference*

Compute the setup cost S_o of the current tableau

Compute the variable cost V_o of the current tableau

Compute the setup cost S_n when the new found non-basic cell come into basis

Compute the variable cost V_n when the new found non-basic cell come into basis

Compute $\Delta Z = (S_n - S_o) + (V_n - V_o)$

IF $\Delta Z < 0$ (reduce objective function value)

Update the tableau by forcing the cell into basis

ELSE

Proceed to Step 4

Step 4:

IF no non-basic cell that can improve the objective function

Proceed to Step 5

ELSE

Loop back to Step 2

Step 5: *Compute the cost and changing the searching direction*

$$O_i = \sum_{j=1}^{J}\sum_{m=1}^{T} S_j * Y_{jm} + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{m=1}^{T}\sum_{n=1}^{T} X_{jkmn} * P_k + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{m=1}^{T}\sum_{n=m}^{T} (n-m) * X_{jkmn} * H_j$$
$$+ \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{m=1}^{T}\sum_{n=1}^{m} (m-n) * X_{jkmn} * B_j$$

IF $i < 4$

$i = i + 1$

Resume the tableau to the structure just after Phase I

Loop back to Step 1

ELSE

Among the O_i, choose the feasible solution with the lowest cost O = $Min(O_i)$, for i = 1,2,3,4

3.3 Design of Computational Experiments

In this section, the design of computational experiments is presented. Since we use the Gilbert and Madan heuristic as the benchmark, we will also follow their computational experiments. The computational study consists of the solutions for 91 (27+27+27) test problems using different parameters. The test problems differ with respect to:

(1) the number of products;

(2) the number of time periods;

(3) the seasonality of demand;

(4) the available regular time capacity and

(5) the setup.

We vary each of these five problem parameters over three values to generate different parameter combinations. The number of products for any problem is 3, 6 or 9. The number of time periods is 12, 24 or 36. The product demands for any problem may either have no seasonality, moderate seasonality or extreme seasonality. The setup cost for each product may be low, medium or high. The level of regular time capacity covers 80%, 100% or 120% of the total demand over the planing horizon. In each problem, the amount of regular time available is constant through out all in the periods over the planning horizon. For convenience the initial inventories and the ending inventories for all products are zero.

3.3.1 Specifications of test problems (3 products and 12 period case)

Let i index problem set, where i = 1 is problem set with no seasonality, i = 2 is problem set with moderate seasonality, and i = 3 is problem set with extreme seasonality.

Product Demands

For problem set i, demand for product j in period n is given by,

$$D_{jn}^{\;i} = \sum_{p=1}^{P} r_p * b_{jn}^{\;i}$$

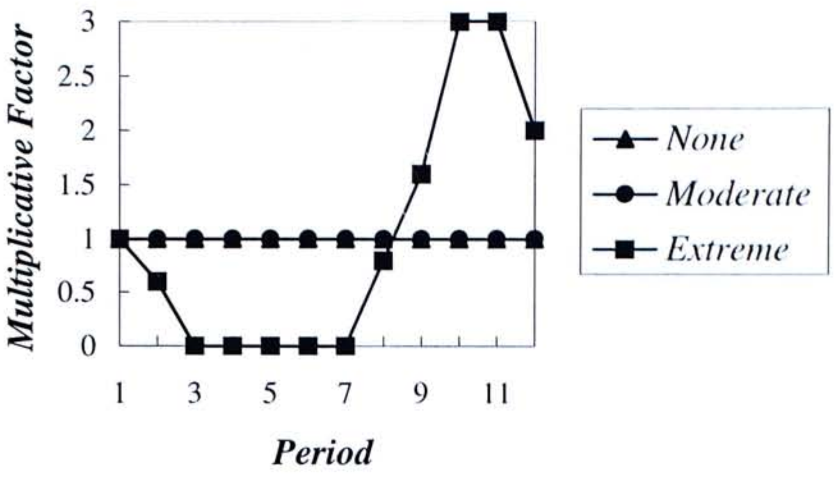
where $b_{jn}^{\;i}$ is a multiplicative seasonality factor (see Figure 3.3 and Figure 3.4) for product j in period n belonging to problem set q, and r_p is the pth random draw from a uniform distribution over the range [u_{dp}, l_{dp}] (see Figure 3.5). P is equal to 5 for products 1 and 2 and P is equal to 10 for product 3.

Period(n)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	
	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	Product 1
None	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	Product 2
(i=1)	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	Product 3
	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	Product 1
Moderate	0.8	0.8	0.7	0.5	0.7	1.0	1.0	1.2	1.3	1.5	1.2	1.0	Product 2
(i=2)	1.0	1.0	1.0	1.2	1.3	1.5	1.3	1.0	0.9	0.7	0.6	0.8	Product 3
	1.0	0.6	0.0	0.0	0.0	0.0	0.0	0.8	1.6	3.0	3.0	2.0	Product 1
Extreme	0.8	0.6	0.3	0.0	0.0	0.6	1.2	1.5	2.0	2.0	1.5	1.2	Product 2
(i=3)	0.3	0.5	0.6	1.0	1.2	1.5	2.0	2.2	1.3	1.0	0.5	0.2	Product 3

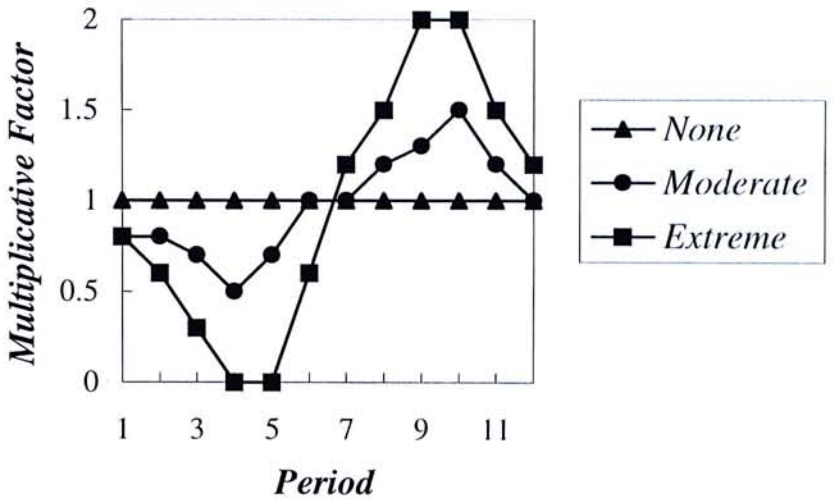
Figure 3.3

Multiplicative seasonality factor

Product 1



Product 2



Product 3

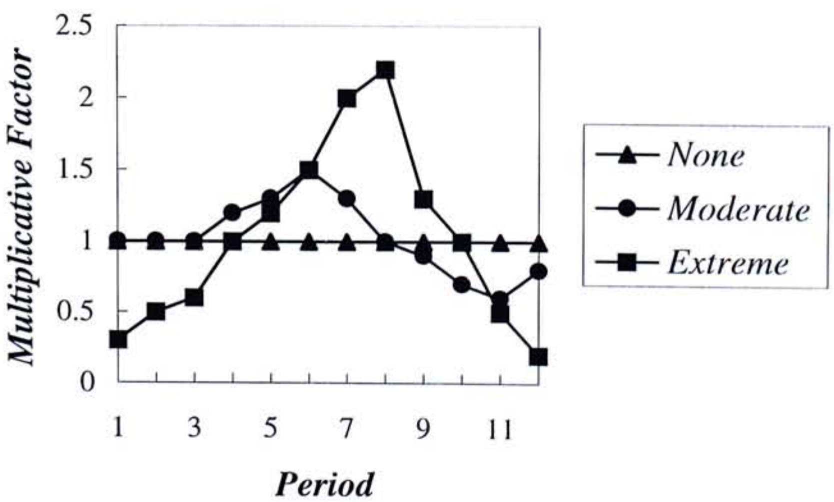


Figure 3.4

Multiplicative seasonality factor of each product shown graphically

Product 1

u_{dp}	20	40	15	25	80
l_{dp}	40	60	25	65	120

u_{sp}	50	50	50	50	50
l_{sp}	150	150	150	150	150

Product 2

u_{dp}	80	15	40	40	55
l_{dp}	120	25	60	60	85

u_{sp}	100	100	100	100	100
l_{sp}	200	200	200	200	200

Product 3

u_{dp}	20	20	30	30	20	30	30	50	60	40
l_{dp}	40	40	50	50	40	70	50	100	80	80

u_{sp}	100	100	100	100	100	100	100	100	100	100
l_{sp}	200	200	200	200	200	200	200	200	200	200

Figure 3.5

Ranges of demands and setup costs for products

Setup Costs

For each product j, the setup cost S_j is given by,

$$S_j = \sum_{p=1}^{P} t_p$$

where t_p is the pth random draw from the a uniform distribution over the range $[u_{sp}, l_{sp}]$ (see Figure 3.2). S_j is the "Medium" setup cost for the product j for all periods for all test problems. To get the "Low" setup cost, we multiply S_j by 0.2 while to get the "High" setup cost, we multiply S_j by 5.

Production costs

The regular time cost is 40.00 per unit and while overtime cost is 60.00 per unit for all periods for all problems

Holding costs

The inventory holding cost H_j per period per unit for all products is given in the Figure 3.6 following,

Product	1	2	3
H_j per period	4	7	6

Figure 3.6

Holding costs per period of products

Backorder costs

The inventory holding cost B_j per period per unit for all products is given by the Figure 3.7 following,

Product	1	2	3
B_j per period	8	14	12

Figure 3.7

Backorder costs per period of products

Available regular time

The available regular time in period T for problem set i is given by,

$$C_{1m}{}^{i} = \sum_{j=1}^{J} \sum_{n=1}^{T} \frac{D_{jn}{}^{i}}{T}$$

where C_{1m} is the available capacity of source type 1. And overtime in period m for problem set i is given by,

$$C_{2m}{}^{i} = 0.25 * C_{1m}{}^{i}$$

where C_{2m} is the available capacity of source type 2.

We multiply $C_{1m}^{\ i}$ by 0.80 for problems with 80% tightness of capacity, by 1.00 for problems with 100% tightness of capacity and by 1.20 for 120% tightness of capacity.

A 6 product problem is two 3 product problems. Similarly a 9 product problem is three 3 product problems.

The setup costs for 6 product and 9 product problems are obtained by replicating the setup costs for a 3 product problem. The holding costs and production costs for 6 product and 9 product problems, are obtained by replicating the costs for a 3 product problem.

Similarly the demands for 6 and 9 product problems are obtained by replicating the demands for a 3 product problem. The capacities are computed exactly like the a 3 product problem.

Moreover, a 24 period problem is two 12 period problems. The demands for a 24 period problem is obtained by replicating 12 period demands. Similarly, a 36 period problem is three 12 period problems. The capacities are computed exactly like a 12 period problem.

3.3.2 Computation of the Lower Bound

We use a Lower Bound procedure presented by Gilbert and Madan (1991) for the purposes of comparing the effectiveness of our algorithm.

A Lower Bound for this problem can be computed by solving two separate relaxations of the problem. In the first relaxation, a Lower Bound on total setup costs is computed. In the second relaxation, a Lower Bound on total variable costs (production cost, holding cost, and backorder cost) is computed.

The Lower Bound on the total setup cost is found by determining a minimum number of setups required for feasibility. For each product, a setup is required for each period in which it is produced. Hence, a Lower Bound on the number of setups required for a particular product is the smallest integer that is greater than the quotient of the total demand for the product and the capacity available per period (i.e. $\left\lceil \dfrac{\sum_{n=1}^{T} D_{jn}}{\sum_{k=1}^{K} C_{kl}} \right\rceil$).

Thus, total Lower Bound of setup costs = $\displaystyle\sum_{j=1}^{J}\left(\left\lceil \frac{\sum_{n=1}^{T} D_{jn}}{\sum_{k=1}^{K} C_{kl}} \right\rceil * S_j\right)$

The Lower Bound on total variable cost is computed by solving the relaxation of the problem with only production, holding, and backorder costs in the objective function. We force the setup cost of products to zero and apply the simplex method on the traditional transportation problem to minimize the objective function value. The Lower Bound on variable cost is computed from the solution.

A Lower Bound for the problem is computed by summing the Lower Bound on total setup cost and the Lower Bound on total production, holding, and backorder cost.

3.4 Computational Results

The heuristic is tested on a variety of test problems. All test problems consist of 3 products and 12 time periods. In addition, the test problems considered three parameters: (1) seasonality of demand, (2) the available regular time capacity, and (3) the setup. Each of the problem parameters were varied over three values to generate 27 problems. The test design utilized is similar to that of Graves (1982) and Gilbert and Madan (1991). The performance of the heuristic is discussed with respect to seasonality in demand, setup costs, and tightness of capacity. The quality of the solutions is evaluated by computing a percentage difference as following:

$$\frac{OFV - LB}{LB} * 100\%$$

where OFV is the objective function value of our heuristic

LB is the Lower Bound value

Ten problem sets each consisting of 27 problems were used for testing the heuristic. The results of the ten problem sets are summarized in Table 3.1. Tables 3.2 through 4 illustrate the effect of the three parameters on the value of the Average Percentage Difference (A.P.D.). In Table 3.2 the A.P.D. between the heuristic solution value and the Lower Bound for different levels of setup for all problems is shown. In Table 3.3 the A.P.D. between the heuristic solution value and the Lower Bound for different levels of setup for all problems across different levels of seasonality is shown. In Table 3.4 the A.P.D. between the heuristic solution value and the Lower Bound for different levels of setup for all problems across different levels of capacity is shown.

The A.P.D. between the heuristic solution value and the Lower Bound is computed for each problem parameter. The average A.P.D. across all 270 problems was 13.96%. The percentage difference ranged from 1.65% to 28.98%.

The most significant effect on A.P.D. is due to the setup cost parameter. As setup cost increases the A.P.D. increases significantly (see Table 3.2). This could be because the Lower Bound procedure relies heavily on the continuous portion of the objective function. As far as seasonality is concerned, the A.P.D. in extreme seasonality cases is lower than those for low and moderate seasonality cases. The A.P.D.s for low and moderate seasonality cases are very similar (see Table 3.3). As far as capacity is concerned, there are minor differences in the A.P.D.s for all three levels (see Table 3.4).

The simultaneous effect of seasonality and setup costs is shown in Table 3.5. For low setup cases and different levels of seasonality the heuristics performs better than medium and high set of cases. In 67% of the cases, the A.P.D. is less than 14.54%. Across all levels of setups, the A.P.D. for extreme seasonality cases was the lowest.

The simultaneous effect of seasonality and level of capacity is shown in Table 3.6. In 67% of the cases, the A.P.D. is less than 15.30%. However, in all cases, the A.P.D. is less than 18.22%. Across all levels of capacity, the A.P.D. for extreme seasonality cases is the lowest.

The simultaneous effect of the level of capacity and setup costs is shown in Table 3.7. For low setup cases and different levels of capacity the heuristics performs better than medium and high set of cases. In 67% of the cases, the A.P.D. is less than 12.83%. For different levels of capacity, the A.P.D. is lowest for the low setup cases.

To investigate the effect of the ratio of backorder cost to holding cost, additional problem sets are tested by setting the ratio of backorder cost and holding cost to 2, 4 and 6. The performance of the heuristic seems to be consistent across all cases (see Table 3.8 to 3.10).

Seasonality	Setup	Capacity	1	2	3	4	5	6	7	8	9	10	A.P.D. (%)
No	Low	80%	3.1	2.8	2.8	2.9	2.9	2.8	2.6	2.8	3.2	3.1	2.83
No	Low	100%	3.8	3.4	3.5	3.5	3.5	3.7	3.3	3.5	4	3.8	3.50
No	Low	120%	4.2	3.7	3.8	5.1	3.9	4.3	3.6	3.8	5.4	4.3	3.98
No	Medium	80%	15.1	13.5	13.8	14.4	13.2	12.9	17.4	16.9	15.5	15.5	13.35
No	Medium	100%	17.6	15.7	23.9	17.6	15	14.3	15.4	15.9	18.2	18.1	15.35
No	Medium	120%	15	14.9	16	15	14.9	16.3	15.7	14.6	17.5	14.6	14.55
No	High	80%	27.9	26.5	27.3	33.1	31.9	25.6	25.7	28.9	44.6	38	28.90
No	High	100%	28.5	33.8	32	28.3	29.5	23.8	32.2	28.5	22.5	28.1	28.52
No	High	120%	28.1	26.9	23.2	27.8	28.2	28.4	26.5	22.2	25.4	24.8	27.01
Moderate	Low	80%	2.9	2.6	2.6	2.7	2.6	2.6	2.4	2.6	3.1	2.8	2.61
Moderate	Low	100%	3.7	3.3	3.3	3.4	3.4	3.6	3.2	3.4	3.9	3.7	3.38
Moderate	Low	120%	4.1	3.7	3.8	4.9	3.9	3.9	3.6	3.8	4.3	4.1	3.84
Moderate	Medium	80%	11	10.9	14.2	10.1	14.6	13.7	13	10.1	15.8	15.3	10.90
Moderate	Medium	100%	15.6	15.3	14.3	14.9	12.9	15	14.4	15.4	16	15.2	14.25
Moderate	Medium	120%	16.4	15.3	14.1	15.9	15.7	13.8	16.4	13.7	16.1	15.4	14.51
Moderate	High	80%	29.5	36.1	30.3	38.4	32.6	34.6	28.6	29.5	30.4	27.1	28.98
Moderate	High	100%	34.4	28.8	30	23.6	21.7	23	25.6	32.6	19.8	30.2	25.30
Moderate	High	120%	29.9	28.4	26.1	31.1	28.9	28.4	31.2	28.3	30.2	36.3	26.48
Extreme	Low	80%	2	1.8	1.9	1.9	2	2	1.8	1.8	2.1	2	1.65
Extreme	Low	100%	2.8	2.6	2.7	2.6	2.7	2.8	2.6	2.6	2.6	2.9	2.56
Extreme	Low	120%	3.2	3	3	3	3.1	3.6	2.9	3	3.4	3.2	2.75
Extreme	Medium	80%	8.5	7	6.7	7.1	6	5.7	5.7	5.5	6.5	6.6	5.93
Extreme	Medium	100%	9.7	9	9.4	9.7	10.2	11.7	10	8.7	9.8	9.3	7.86
Extreme	Medium	120%	9.6	9.6	9.2	12.2	10.5	13.2	10.3	10.3	11.3	10.0	9.88
Extreme	High	80%	11.5	14.6	13.2	12.2	13.3	10.1	11.7	13.7	11.7	15.4	12.92
Extreme	High	100%	19.2	21.1	22.5	22.5	25.4	24.5	22.3	16.8	23.8	23	19.07
Extreme	High	120%	29.6	25.2	30.7	26.7	30.2	32.4	28.8	30.4	35	23.5	24.85

Table 3.1 Average Percentage Difference of the 270 problem cases

	Setup Cost		
	Low	*Medium*	*High*
A.P.D.	3.16	12.22	26.55

Table 3.2

The effect of setup costs on A.P.D.

	Seasonality		
	None	*Moderate*	*Extreme*
A.P.D.	15.89	15.28	10.76

Table 3.3

The effect of seasonality on A.P.D.

	Capacity		
	80%	*100%*	*120%*
A.P.D.	14.41	13.26	14.26

Table 3.4

The effect of tightness of capacity on A.P.D.

Setup	**Seasonality**		
	None	*Moderate*	*Extreme*
Low	3.34	3.41	2.73
Medium	14.54	13.23	8.89
High	29.78	29.21	20.66

Table 3.5

The effect of seasonality and setup costs on A.P.D.

Capacity	**Seasonality**		
	None	*Moderate*	*Extreme*
80%	18.22	16.38	8.64
100%	14.14	15.43	10.21
120%	15.30	14.03	13.43

Table 3.6

The effect of seasonality and tightness of capacity on A.P.D.

Capacity	**Setup**		
	Low	*Medium*	*High*
80%	2.72	12.19	28.32
100%	3.09	11.64	25.06
120%	3.67	12.83	26.27

Table 3.7

The effect of setup costs and tightness of capacity on A.P.D.

	Setup Cost		
	Low	*Medium*	*High*
A.P.D.	3.42	12.29	25.64

	Seasonality		
	None	*Moderate*	*Extreme*
A.P.D.	14.82	15.85	10.68

	Capacity		
	80%	*100%*	*120%*
A.P.D.	12.52	13.44	15.38

Setup	**Seasonality**		
	None	*Moderate*	*Extreme*
Low	4.02	3.82	2.42
Medium	15.62	13.08	8.17
High	24.82	30.65	21.45

Capacity	**Seasonality**		
	None	*Moderate*	*Extreme*
80%	14.90	15.42	7.25
100%	14.97	15.05	10.30
120%	14.58	17.08	14.48

Capacity	**Setup**		
	Low	*Medium*	*High*
80%	2.77	9.90	24.90
100%	3.52	13.42	23.38
120%	3.97	13.55	28.63

Table 3.8

The effect of ratio of backorder cost to holding cost ($B_j = 2H_j$)

	Setup Cost		
	Low	*Medium*	*High*
A.P.D.	3.61	12.87	31.20

	Seasonality		
	None	*Moderate*	*Extreme*
A.P.D.	18.30	17.77	11.61

	Capacity		
	80%	*100%*	*120%*
A.P.D.	15.15	15.42	17.12

Setup	Seasonality		
	None	*Moderate*	*Extreme*
Low	3.60	4.48	2.75
Medium	15.10	14.95	8.57
High	36.20	33.88	23.52

Capacity	Seasonality		
	None	*Moderate*	*Extreme*
80%	19.03	17.98	8.43
100%	18.30	16.88	11.07
120%	17.57	18.45	15.33

Capacity	Setup		
	Low	*Medium*	*High*
80%	2.95	11.68	30.82
100%	3.80	13.27	29.18
120%	4.08	13.67	33.60

Table 3.9

The effect of ratio of backorder cost to holding cost ($B_j = 4H_j$)

	Setup Cost		
	Low	*Medium*	*High*
A.P.D.	3.19	13.10	33.09

	Seasonality		
	None	*Moderate*	*Extreme*
A.P.D.	18.72	18.70	11.97

	Capacity		
	80%	*100%*	*120%*
A.P.D.	15.43	16.46	17.49

Setup	Seasonality		
	None	*Moderate*	*Extreme*
Low	3.27	3.77	2.55
Medium	15.58	15.45	8.27
High	37.30	36.88	25.08

Capacity	Seasonality		
	None	*Moderate*	*Extreme*
80%	19.75	17.60	8.95
100%	17.73	20.28	11.35
120%	18.67	18.22	15.60

Capacity	Setup		
	Low	*Medium*	*High*
80%	2.12	11.95	32.23
100%	3.67	13.72	31.98
120%	3.80	13.63	35.05

Table 3.10

The effect of ratio of backorder cost to holding cost ($B_j = 6H_j$)

3.5 Comparison to Millar and Yang's Algorithm

Millar and Yang (1994) develop a solution algorithm using Lagrangain decomposition and Langrangain relaxation techniques. Their formulation considers the backordering but fails to consider more than one type of capacity sources. Since their formulation does not consider the overtime capacity source, we need to modify our problem to make a comparison with their algorithm possible. In our formulation, we disallow overtime production by setting overtime capacity to zero for each period.

In their computational experiments, they only consider the three type of parameters, setup cost (high and low), capacity tightness (low, medium and high) and backorder cost (low, medium and high). These 6*3 generated problems are tested by their Lagrangain decomposition and relaxation, and compared to the lower bound. They measure the quality of the solution by computing a gap value as follows:

$$\text{Gap} = 200 * \frac{\text{Solution Value - Lower Bound}}{\text{Solution Value + Lower Bound}}$$

3.5.1 Comparison Results

The results are shown in the following tables (Table 3.11 to 3.13). We put an asterisk '*' to mark a case in which that algorithm performs better than the other two.

Setup	Capacity	Our Algorithm	Decomposition	Relaxation
Low	Low	3.21*	19.29	18.80
Low	Medium	3.51*	22.50	22.95
Low	High	4.21*	12.46	13.02
Medium	Low	9.82*	12.99	12.57
Medium	Medium	11.06*	11.18	21.97
Medium	High	13.88	7.22*	10.44

Table 3.11

Gap value ($B_j = 2H_j$)

Setup	Capacity	Our Algorithm	Decomposition	Relaxation
Low	Low	3.20*	7.74	8.32
Low	Medium	3.48*	7.40	8.13
Low	High	4.18*	8.34	8.64
Medium	Low	10.39	7.14*	13.09
Medium	Medium	11.24	6.31*	10.11
Medium	High	13.90	5.74*	9.77

Table 3.12

Gap value ($B_j = 4H_j$)

Setup	Capacity	Our Algorithm	Decomposition	Relaxation
Low	Low	3.26	2.74	2.44*
Low	Medium	3.57*	8.10	5.70
Low	High	4.28*	8.18	7.16
Medium	Low	10.96	6.67*	11.50
Medium	Medium	11.60	8.05*	12.82
Medium	High	14.03	6.81*	7.66

Table 3.13

Gap value ($B_j = 6H_j$)

When backorder costs are low (Backorder cost is double of holding cost, $B_j = 2H_j$), our algorithm performs much better than the Lagrangian Decomposition and Relaxation (see Table 3.11).

In 50% of the cases, our algorithm performs better than Lagrangian Decomposition and Relaxation when the setup cost is low and backorder cost is medium (see Table 3.12).

When backorder cost increases ($B_j = 6H_j$), our algorithm performs better than the Lagrangian Decomposition in 33% of the cases. In 67% of cases, the Gap value of our algorithm is less than those of Lagrangian Relaxation (see Table 3.13).

3.6 Conclusion

In this chapter, we formulated the SLCR problem with backordering as a fixed charge transportation problem. We also computed the Lower Bound for comparison. This formulation has the advantage of considering both regular time and overtime. The algorithm performed quite well when compared to the Lower Bound (which is infeasible in some of the case). Moreover, we compared the performance with the results by the Millar and Yang's algorithm. Our algorithm performed quite well when the setup cost is low.

Chapter 4

The Optimization Based Algorithm

In this chapter, we propose an efficient optimisation based algorithm for Single Level Constrained Resource (SLCR) problem which disallows backordering. Backordering is disallowed in situations where backorder cost is extremely high or where failure to deliver products on time means loss of constracts.

As discussed in Chapter 2, several formulations have been used in the literature. The formulation proposed by Gilbert and Madan (1991) is adopted in this research because most existing formulations ignore alternate sources of production capacity. In today's highly competitive marketing environment, manufacturers have to emphasize on order winners such as flexibility, availability, and delivery, in addition to cost and quality. Overtime capacity helps manufacturers achieve the desirable flexibility, availability and delivery performance.

In order to solve the problem efficiently, solution algorithms often use a heuristic exploiting the special structure of the problem. The Gilbert and Madan formulation carries a structure similar to a type of fixed charge transportation problem. In the fixed charge transportation problem, each cell in the transportation tableau is assumed to have fixed cost and variable cost. A fixed charge is shared by a group of cells in the transportation tableau. Therefore, existing algorithms for the fixed charge transportation problem are not designed to exploit the special structure of the SLCR problem discussed in the thesis.

Similar to the algorithm for the backordering case in the previous chapter, our algorithm uses the heuristic based on a primal simplex algorithm for the transportation problem. In the algorithm, the pivot rule chooses the non-basic variable to enter the basis to bring the largest decrease in the objective function value. The algorithm terminates if there is no more improvement by bringing a non-basic variable into the basis. Computational results show that our algorithm outperforms the Gilbert and Madan algorithm in terms of the quality of solutions generated.

4.1 The Formulation

Our formulation of the SLCR problem without backordering is the same as the formulation presented by Gilbert and Madan (1991). The planning horizon is divided into T equal time periods. There are J different products. The demand D_{jn}, $j = 1,2,...,J$ and $n = 1,2,...,T$ for product j is required in period n.

A product may be produced using K different types of capacity (e.g. regular time, and overtime). C_{km}, $k = 1,2,...,K$; $m = 1,2,...,T$ denotes the number of units of capacity type k available in period m.

We assume that the number of units of capacity used in producing a given product is directly proportional to the quantity produced. Therefore, we express all quantities in units of capacity required. For product j produced in period m using capacity type k, the demand, the production level and the inventory, will be expressed in units of capacity. We assume that, one unit of a product is produced by one unit of capacity for simplicity. This assumption is not restrictive. We can easily modify our formulation for the situation where one unit of capacity may produce several units of a product.

The cost of capacity type k, $k = 1,2,...,K$, is P_k dollars per unit, $P_1 \leq P_2 \leq ... \leq P_k$. To be consistent with the Gilbert and Madan formulation, we disallow backordering. Therefore, the proposed formulation does not allow products be produced to meet the demands of earlier periods.

In other words, products produced during a given period can only be used to meet the demand for the product during that period or in the subsequent periods. Each product j, $j = 1,2,...,J$, has a holding cost of H_j dollars per period. Total holding cost of a period is based on the ending inventory of the period.

Each product requires a setup for each period in which it is produced. The cost of a setup for product j is S_j dollars per setup, j = 1,2,...,J. It is assumed that downtime consumed by the setup operation is negligible.

Using the following notations, the SLCR problem can be formulated as a type of fixed charge transportation problem. Let j and k index product and type of production capacity respectively, where j = 1,2,...,J ; k = 1,2,...,K. Let m index period in which product is produced and n index period in which product is demanded, where m = 1,2,...,T ; n = 1,2,...,T.

Let,

J = number of products;

K = number of type of production capacity available;

T = number of time periods in the planing horizon;

D_{jn} = demand of product j in period n;

C_{km} = units of capacity type k available in period m;

S_j = setup cost for product j;

H_j = holding cost in dollars per period of product j;

P_k = production cost per unit of capacity type k;

Y_{jm} = a binary variable that indicates the presence or absence of a setup for product j in period m.

Further, we let X_{jkmn} be the quantity of product j produced using production capacity source k in period m to meet the demand in period n.

The fixed charge transportation formulation of the Single Level Constrained Resource problem is given below:

Minimize :

$$\sum_{j=1}^{J}\sum_{m=1}^{T} S_j * Y_{jm} + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{m=1}^{T}\sum_{n=m}^{T} X_{jkmn} * P_k + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{m=1}^{T}\sum_{n=m}^{T} (n-m) * X_{jkmn} * H_j \quad (4.1)$$

Subject to :

$$\sum_{k=1}^{K}\sum_{m=1}^{n} X_{jkmn} = D_{jn} \qquad j = 1,2,\ldots,J \quad n = 1,2,\ldots,T \qquad (4.2)$$

$$\sum_{j=1}^{J}\sum_{n=m}^{T} X_{jkmn} \le C_{km} \qquad k = 1,2,\ldots,K \quad m = 1,2,\ldots,T \qquad (4.3)$$

$$Y_{jm} = \begin{cases} 1 \text{ if } \sum_{k=1}^{K}\sum_{n=m}^{T} X_{jkmn} > 0 \\ 0 \text{ otherwise} \end{cases} \qquad j = 1,2,\ldots,J \quad m = 1,2,\ldots,T \qquad (4.4)$$

$$X_{jkmn} \ge 0 \qquad j = 1,2,\ldots,J \quad k = 1,2,\ldots,K \quad m = 1,2,\ldots,T \quad n = 1,2,\ldots,T \qquad (4.5)$$

Problem (4.1) to (4.5) is a variation of the fixed charge transportation problem. A group of cells in the transportation tableau shares a fixed charge. This feature makes our formulation different from the traditional fixed charge transportation problem.

Like a transportation problem, (4.1) to (4.5) can be represented by a transportation tableau, in which rows are supplies (production capacities) and columns are demands. It turns out that the transportation tableau is very helpful in designing our solution algorithm. In addition, we feel that the transportation tableau is also a good decision tool. With the transportation tableau, we can easily explain a production schedule to plant managers and workers.

The objective function (4.1) is the minimization of total cost. The first term in the objective function represents the sum of setup costs for all products in every period, the second term represents total production capacity cost, and the third term represents total inventory holding cost.

Constraints (4.2) are demand constraints which require that demand, D_{jn}, of product j, $j = 1,2,...,J$, during period n, $n = 1,2,...,T$, is met by X_{jkmn}, $m \leq n$. The sum of production quantity of product j in period m (m is from 1 to n) is used to meet the demand of period n.

Constraints (4.3) are capacity constraints which require that $\sum_{j=1}^{J}\sum_{n=m}^{T} X_{jkmn}$, the total amount of production capacity of type k used in period n (n is from m to T) to produce different products, does not exceed the total amount of capacity of type k available in period m, $m = 1,2,...,T$; $k = 1,2,...,K$. It should be noted that constraints (3.2) and (3.3) exclude backorder possibility.

Usually the total demand and the total supply are equal. The problem is infeasible if the total demand is greater than the total supply. However, in cases without backordering, we have additional information, i.e. the problem is infeasible if total demand for a given period is greater than total available capacity of the given period. If the total demand is less than the supply, then an additional column (dummy column) with demand equaling excess supply is added.

If total demand for products over the planning periods is less than the total available capacity, we define $D_{J+1\ T+1}$ as follows:

$$D_{J+1T+1} = \sum_{k=1}^{K}\sum_{m=1}^{T} C_{km} - \sum_{j=1}^{J}\sum_{n=1}^{T} D_{jn}$$

where $D_{J+1\ T+1}$ is demand for the dummy column ($J+1$, $T+1$) (destination). We then add the following demand constraint for dummy demand.

$$\sum_{k=1}^{K}\sum_{m=1}^{T} X_{J+1\,km\,T+1} = D_{J+1T+1}$$

Constraint (4.3) will be changed to equality constraints as shown below:

$$\sum_{j=1}^{J}\sum_{n=m}^{T} X_{jkmn} + X_{J+1\,km\,T+1} = C_{km} \qquad k = 1,2,\ldots,K \quad m = 1,2,\ldots,T$$

Constraints (4.4) are logical constraints which account for setups for product j in period m. A setup cost S_j is incurred once for product j in period m if the product is produced in period m (using any type of capacity). In other words, if $\sum_{k=1}^{K}\sum_{n=m}^{T} X_{jkmn}$ is positive, $j = 1,2,\ldots,J$; $m = 1,2,\ldots,T$, then Y_{jm} is One.

Constraints (4.5) are non-negativity constraints.

Figure 4.1 is a transportation tableau associated with the formulated problem. A supply in the transportation tableau is denoted by C_{km} which is the quantity of source k production capacity available in period m. A demand is denoted by D_{jn} which is the quantity of product j demanded in period n.

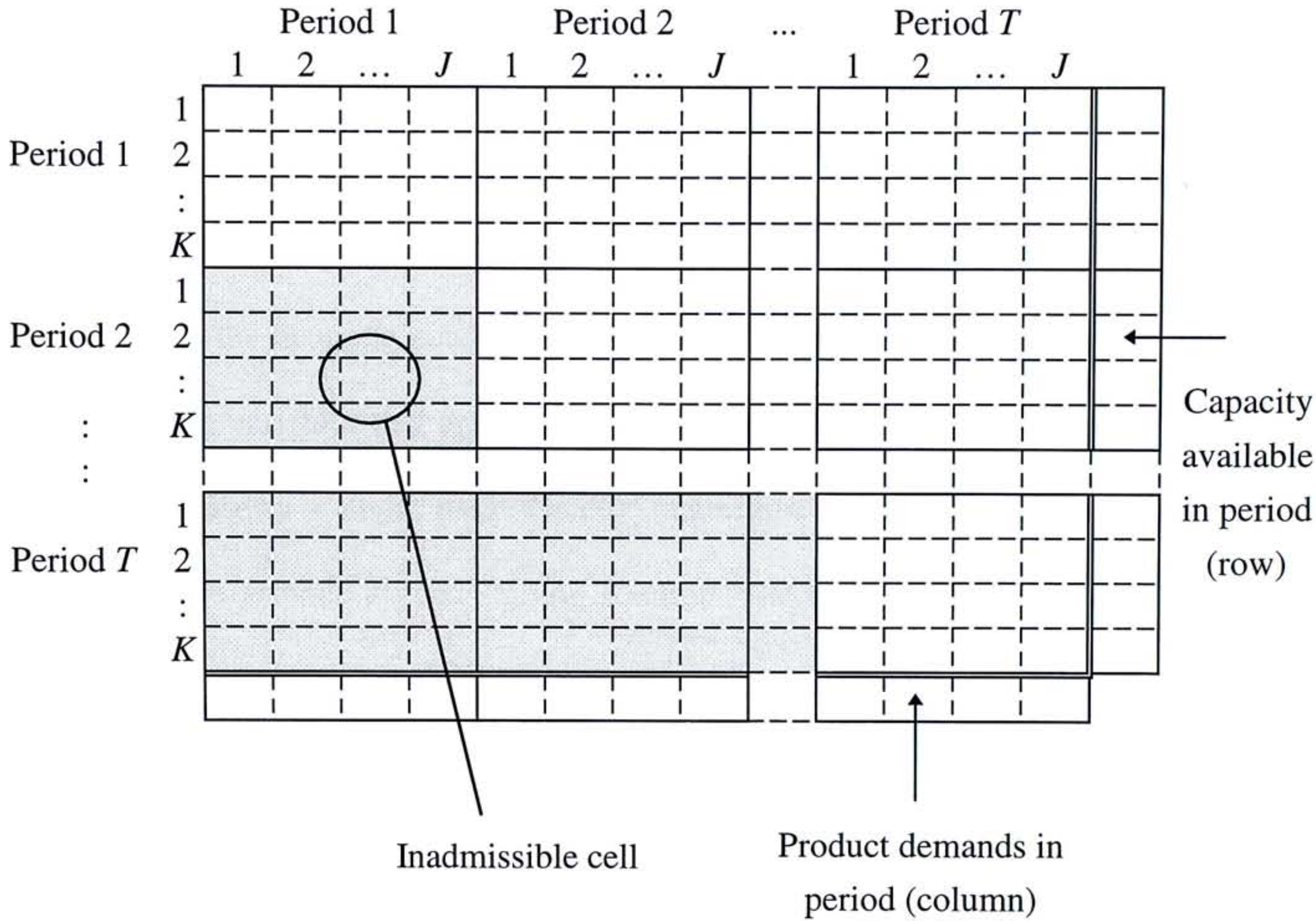


Figure 4.1

An example of transportation tableau without backordering

The cell in row (k,m) and column (j,n) is denoted as cell $((k,m)\ (j,n))$. The flow in this cell corresponds to the variable X_{jkmn}. The value of the setup variable Y_{jm} is implicitly determined by constraint (3.4). Hence a feasible tableau solution corresponds to a feasible solution to problem (3.1) to (3.5). Since backordering is not allowed, any cell $((k,m)\ (j,n))$ with $m > n$ is infeasible and therefore inadmissible.

4.2 The Algorithm

In this section, the heuristic for solving the SLCR problem without backordering will be presented. The heuristic works with basic feasible solutions to the transportation problem (4.1) to (4.5). Similar to the Gilbert and Madan (1991) algorithm, the heuristic consists of three phases. The first phase introduces into the basis some variables that are required to meet the demands of the first period. The second phase is a single pass "Greedy" algorithm that selects the remaining basic variables to provide a good starting solution. The third phase makes an attempt to improve the solution obtained in the second phase by replacing variables in the basis.

Our heuristic is better in the sense that we develop a better approximation of cost per unit of production. This helps us in finding a better initial solution. In addition, we develop better search procedures to identify the most desirable non-basic candidate into the basic solution. The search procedures remove the inherited bias in the Gilbert and Madan search procedures.

4.2.1 Phase I

In this phase, we must schedule production using different capacity types to meet the demands for all products in the first period. If D_{j1} is positive, then a feasible solution should have the production using one type or a combination of types of capacity for product j scheduled in the first period. The demands for the products in the first period must be met by using the least expensive type of capacity available in the first period, although there may be more than one type of capacity available.

The output from this phase is either the assignment of values to X_{jk11}, j = 1,2,...,J ; k = 1,2,...,K, such that $\sum_{k=1}^{K} X_{jk11} = D_{j1}$, or the indication that the problem is infeasible. The infeasible case occurs when total supply from all capacity types in the first period is not enough to meet total demand of first period.

In Phase I and Phase II of our heuristic, we let $\overline{D_{jn}} = D_{jn} - \sum_{k=1}^{K}\sum_{m=1}^{n} X_{jkmn}$ be the remaining unmet demand for product j in period n, for j = 1,2,...,J ; n = 1,2,...,T. Similarly let $\overline{C_{km}} = C_{km} - \sum_{j=1}^{J}\sum_{n=m}^{T} X_{jkmn}$ be the unused capacity of type k in period m, for k = 1,2,...,K ; m = 1,2,...,T. The detailed steps in Phase I are shown in the following.

Step 1: *Initialization*

Initialize $X_{jkmn} = 0, \overline{C_{km}} = C_{km}, \overline{D_{jn}} = D_{jn}$

for j = 1,2,...,J ; k = 1,2,...,K ; m = 1,2,...,T ; n = 1,2,...,T

Set j = 1, k = 1

Step 2: *Satisfy product demand in the first period*

IF $k > K$ (When all the capacities in the first period have been used up, and there is remaining unmet demand for any product in period 1)

Terminate with an infeasible condition;

ELSE IF $j > J$ (When all product demands of the first period has been fulfilled)

Proceed to Phase II;

ELSE

Assign value to variable associated with cell $((k,1)\ (j,1))$ by

$$X_{jk11} = Minimum(\overline{D_{j1}}, \overline{C_{k1}})$$

Step 3: *Update demand and capacity*

IF $\overline{D_{j1}} \leq \overline{C_{k1}}$

$$\overline{C_{k1}} = \overline{C_{k1}} - \overline{D_{j1}}$$

$$\overline{D_{j1}} = 0$$

Mark Column $(j,1)$ (To indicate that the corresponding demand has been met)

$j = j + 1$

ELSE

$$\overline{D_{j1}} = \overline{D_{j1}} - \overline{C_{k1}}$$

$$\overline{C_{k1}} = 0$$

Mark Row $(k,1)$ (To indicate that the corresponding capacity has been completely used)

$k = k + 1$

Loop back to Step 2

4.2.2 Phase II

In Phase II, we may use any procedures for constructing an inital solution. In this research we use the procedure similar to the Vogel's Approximation Method (Reinfeld and Vogel 1958).

This phase is similar to the Vogel's Approximation Method except that the cost coefficients consider fixed charge associated with each cell in the tableau. In our problem, a group of cells share a fixed charge (see Figure 4.2). The costs of all the cells which share the same fixed charge will be updated, whenever one of these cells becomes non-zero.

According to Vogel's Approximation Method, penalty is calculated to indicate where departure from lowest cost allocations will bring the highest increase in cost. Instead of finding the penalty using the real production costs, we use the nominal cost (which takes into account the fixed cost) to calculate the penalty. The maximum possible value is assigned to the variable associated with the cell having the largest penalty.

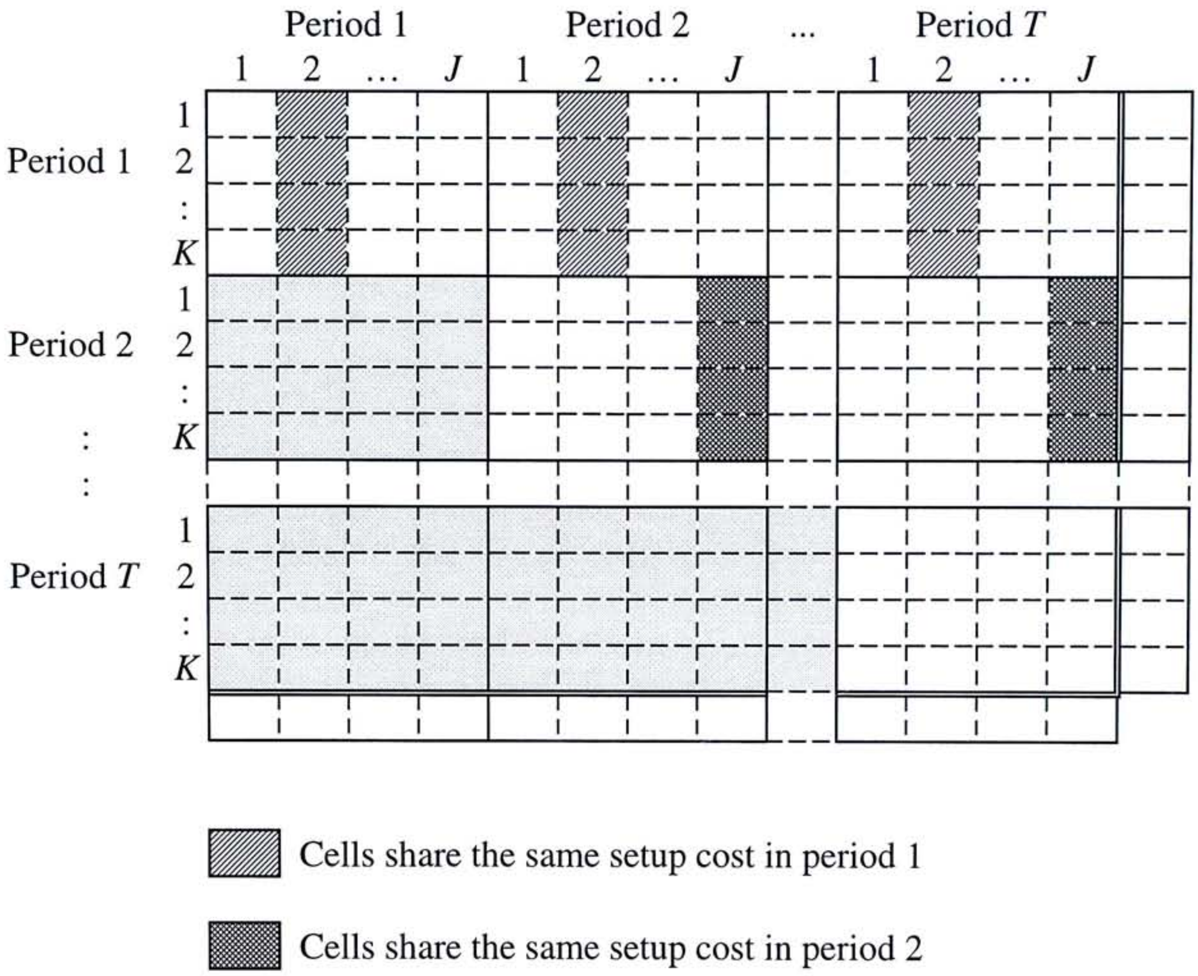


Figure 4.2

Setup cost shared by cells

If cell $((k,m)\ (j,n))$ is chosen to enter the basis, X_{jkmn} will take on a value equal to minimum of $\overline{D_{jn}}$ or $\overline{C_{km}}$. If a setup has already been done for the product j in period m, the objective function will increase by:

$X_{jkmn} * (P_k + (n-m) * H_j)$

On the other hand if the setup has not been done for product j in period m then the objective function will increase by:

$X_{jkmn} * (P_k + (n-m) * H_j) + S_j$

There are other cells sharing the same setup in each period m, $m = 1,2,...,T$, using different types of capacity k , $k = 1,2,...,K$. We may have more than one non-zero cell to share the setup within the same period m. This reduces the average increase setup cost per unit. Producing product j in period m to meet the demand for period n ($n > m$) bears holding cost per unit by $(n\text{-}m) * H_j$.

We introduce a new term w, holding period window size. Each time we only consider the situation that the products produced in period m will be held for w period(s) to meet the demand, where $w = 0,1,...,T\text{-}1$. Then there will be $k * (w+1)$ cells sharing the same setup cost (see Figure 4.2) and the total demand associated to these cells for product j will be $\sum_{t=m}^{Min(m+w,T)} \overline{D_{jt}}$, $j = 1,2,\dots,J$.

Similar to the demand, the total capacity used by the cells which share the same setup is $\sum_{q=1}^{K} \overline{C_{qm}}$. Then the average rate of increase in the objective function per unit of product j met in period n is:

$$P_k + (n-m) * H_j + \frac{S_j}{Min(\sum_{t=m}^{Min(m+w,T)} \overline{D_{jt}}, \sum_{q=1}^{K} \overline{C_{qm}})} \qquad w = 0,1,...,T\text{-}1$$

It is unreasonable to fix the holding period window size to a value between 0 and T-1. We repeat phase II by T times to get the best solution (The best one is the feasible solution that has the lowest the objective function value). The detailed steps of Phase II are shown in the following.

Step 1: *Initialize window size*

Initialize $w = 0$

Step 2: *Computation of nominal cost for cells in tableau*

Compute the nominal cost R_{jkmn} for admissiable cells $((k,m)\ (j,n))$

(i.e. cells with $m \leq n$).

$$R_{jkmn} = P_k + (n - m) * H_j \qquad j = 1,2,\ldots,J$$

$$\text{if } \sum_{q=1}^{K}\sum_{t=m}^{T} X_{jqmt} > 0 \qquad k = 1,2,\ldots,K$$

$$R_{jkmn} = P_k + (n-m) * H_j + \frac{S_j}{Min(\sum_{t=m}^{Min(m+w,T)} \overline{D_{jt}}, \sum_{q=1}^{K} \overline{C_{qm}})} \qquad m = 2,3,\ldots,T \quad n = 1,2,\ldots,T$$

$$\text{if } Min(\sum_{t=m}^{Min(m+w,T)} \overline{D_{jt}}, \sum_{q=1}^{K} \overline{C_{qm}}) > 0$$

$$\text{AND } \sum_{q=1}^{K}\sum_{t=m}^{T} X_{jqmt} = 0$$

For inadmissible cells (i.e. cells with $m > n$), define the cost as M, where M is a very large number.

Step 3: *Computation of row penalty*

For each unmarked row compute penalty E_{km}

where E_{km} is the value difference between the lowest nominal cost and the second lowest nominal cost in the row (k,m)

Step 4: *Computation of column penalty*

For each unmarked column compute penalty G_{jn}

where G_{jn} is the value difference between the lowest nominal cost and the second lowest nominal cost in the column (j,n)

Step 5: *Selection of cell to enter into basis*

Let $E_{k'm'} = Max(E_{km})$

and $G_{j'n'} = Max(G_{jn})$

IF $E_{k'm'} > G_{j'n'}$

$R_{j'k'm'n'} = Min(R_{jk'm'n}) \qquad j = 1,2,...,J\ ;\ n = 1,2,...,T$

ELSE

$R_{j'k'm'n'} = Min(R_{j'kmn'}) \qquad k = 1,2,...,K\ ;\ m = 1,2,...,T$

where $((k',m')\ (j',n'))$ is the cell selected to enter the basis.

Step 6: *Assign value to cell* $((k',m')\ (j',n'))$

$$X_{j'k'm'n'} = Minimum(\overline{D_{j'n'}}\ ,\ \overline{C_{k'm'}})$$

IF $X_{j'k'm'n'} = 0$

Mark column (j',n') and loop back to Step 2

ELSE

Proceed to Step 7

Step 7: *Update row capacity and column demand*

IF $\overline{D_{j'n'}} \leq \overline{C_{k'm'}}$

$$\overline{C_{k'm'}} = \overline{C_{k'm'}} - \overline{D_{j'n'}}$$

$$\overline{D_{j'n'}} = 0$$

Mark Column (j',n') (To indicate that the corresponding demand has been met)

ELSE

$$\overline{D_{j'n'}} = \overline{D_{j'n'}} - \overline{C_{k'm'}}$$

$$\overline{C_{k'm'}} = 0$$

Mark Row (k',m') (To indicate that the corresponding capacity has been completely used)

Step 8: *Re-compute of the nominal costs*

IF any column or row are unmarked (means demands are not met)

Re-compute the nominal costs for cells which share a fixed charge with the cell $((k',m')\ (j',n'))$.

$R_{j'km'n} = P_k + (n-m') * H_{j'}$ $\qquad k = 1,2,\ldots,K\,;\ n = m', m'+1,\ldots,T$

Loop back to Step 2

ELSE

Proceed to Step 9

Step 9: *Repeat with other window sizes*

$$TC_w = \sum_{j=1}^{J}\sum_{m=1}^{T} S_j * Y_{jm} + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{m=1}^{T}\sum_{n=m}^{T} X_{jkmn} * P_k + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{m=1}^{T}\sum_{n=m}^{T} (n-m) * X_{jkmn} * H_j$$

IF $w < T$-2

$w = w + 1$

Unmark all the rows and columns except rows and columns marked in the Phase I.

Loop back to Step 2

ELSE

Among the TC_w, choose the feasible solution with the lowest cost $TC = Min(TC_w)$

and proceed to Phase III

4.2.3 Phase III

Phase III uses a heuristic to improve the solution obtained in Phase II. The algorithm is similar to a primal network simplex algorithm but it takes into consideration fixed costs as well as variable costs in pricing pivots.

The formulated SLCR problem is a variation of the fixed charge transportation problem. If a non-basic variable is introduced into the basis, the total variable cost and the total fixed cost of the solution may be affected. The total fixed cost (setup cost) incurred may be affected because of the following cases:

(1) The setup cost associated with variables which turn positive from zero;

(2) The setup cost associated with variables which turn zero from a positive value.

In Phase III, the non-basic variable whose entrance into the basis will bring the largest reduction in total cost will be selected. For a basic feasible solution if a non-basic variable X_{jkmn} enters the basis, the resulting change in the objective function value is given by ΔZ,

$$\Delta Z = \Delta S + \Delta V$$

where ΔS is the setup cost difference when a new non-basic cell enters basis;

ΔV is the variable cost difference when a new non-basic cell enters basis.

In searching for a better solution, we use the four direction search approach discussed in the previous chapter. The detailed steps in Phase III shown as follows.

Step 1: *Initialize the searching direction i (i = 1,2,3,4)*

1 = Forward Across,

2 = Forward Down,

3 = Backward Across and

4 = Backward Up

Initialize $i = 1$

Step 2: *Search for non-basic cells*

Find the next non-basic cell in the tableau by following the direction i

Step 3: *Calculation of the difference*

Compute the setup cost S_o of the current tableau

Compute the variable cost V_o of the current tableau

Compute the setup cost S_n when the newly found non-basic cell come into basis

Compute the variable cost V_n when the newly found non-basic cell come into basis

Compute $\Delta Z = (S_n - S_o) + (V_n - V_o)$

IF $\Delta Z < 0$ (reduce objective function value)

Update the tableau by introducing the cell into basis

ELSE

Proceed to Step 4

Step 4: *Check whether there is room for improvement*

IF no non-basic cell that can improve the objective function

Proceed to Step 5

ELSE

Loop back to Step 2

Step 5: *Compute the cost and changing the searching direction*

$$O_i = \sum_{j=1}^{J}\sum_{m=1}^{T} S_j * Y_{jm} + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{m=1}^{T}\sum_{n=m}^{T} X_{jkmn} * P_k + \sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{m=1}^{T}\sum_{n=m}^{T} (n-m) * X_{jkmn} * H_j$$

IF $i < 4$

$i = i + 1$

Resume the tableau to the structure just after Phase II

Loop back to Step 2

ELSE

Among the O_i, choose the feasible solution with the lowest cost $O = Min(O_i)$, for $i = 1,2,3,4$

4.3 An Illustrative Example

Consider a simple example with the following characteristics:

Number of Periods (T)	4
Number of Products (J)	2
Kind of capacity sources (K)	2 (Regular Time & Over Time)
Setup cost of Product 1 (S_1)	\$ 396
Setup cost of Product 2 (S_2)	\$ 543
Holding cost of Product 1 (H_1)	\$ 4 per period
Holding cost of Product 2 (H_2)	\$ 7 per period
Backordering is not allowed	

The transportation tableau for the example is given in Figure 4.3. The first assignment is made in Figure 4.4. The initial solution is obtained after the 16th iteration in Phase II (see Figure 4.5).

$P_1 + H_1 = 40 + 4 = 44$

Period	n=1		n=2		n=3		n=4			C_{km}
	j=1	j=2	j=1	j=2	j=1	j=2	j=1	j=2		
m=1 k=1	40	40	44	47	48	54	52	61	M	530
k=2	60	60	64	67	68	74	72	81	M	133
m=2 k=1	M	M	40	40	44	47	48	54	M	530
k=2	M	M	60	60	64	67	68	74	M	133
m=3 k=1	M	M	M	M	40	40	44	47	M	530
k=2	M	M	M	M	60	60	64	67	M	133
m=4 k=1	M	M	M	M	M	M	40	40	M	530
k=2	M	M	M	M	M	M	60	60	M	133
D_{jn}	258	274	215	275	256	272	252	316	534	

M : a very large number

Dummy column
(Total Capacities > Total Demands)

Inadmissible cell

Figure 4.3

Initial tableau of the problem example

1st assignment of Phase I

530 - 274 = 256

Period	n=1		n=2		n=3		n=4			C_{km}
	j=1	j=2	j=1	j=2	j=1	j=2	j=1	j=2		
m=1 k=1	42	42 274	46	49	50	56	54	63	M	256
k=2	63	64	67	71	71	78	75	85	M	133
m=2 k=1	M	M	42	42	46	49	50	56	M	530
k=2	M	M	63	64	67	71	71	78	M	133
m=3 k=1	M	M	M	M	42	42	46	49	M	530
k=2	M	M	M	M	63	64	67	71	M	133
m=4 k=1	M	M	M	M	M	M	42	42	M	530
k=2	M	M	M	M	M	M	63	64	M	133
D_{jn}	258	0	215	275	256	272	252	316	534	

Demand have been meet and the column is marked

Nominal cost : 40 + 543/316 = 41.6 = 42

Figure 4.4

Tableau after First assignment

Period	n=1		n=2		n=3		n=4			C_{km}
	j=1	j=2	j=1	j=2	j=1	j=2	j=1	j=2		
m=1 k=1	$\boxed{40}$ 256	$\boxed{40}$ 274	$\boxed{44}$	$\boxed{47}$	$\boxed{48}$	$\boxed{54}$	$\boxed{52}$	$\boxed{61}$	$\boxed{M}$	0
k=2	$\boxed{60}$ 2	$\boxed{60}$	$\boxed{64}$	$\boxed{77}$	$\boxed{68}$ 131	$\boxed{84}$	$\boxed{74}$	$\boxed{91}$	$\boxed{M}$	0
m=2 k=1	$\boxed{M}$	$\boxed{M}$	$\boxed{40}$ 215	$\boxed{40}$ 275	$\boxed{44}$ 40	$\boxed{47}$	$\boxed{48}$	$\boxed{54}$	$\boxed{M}$	0
k=2	$\boxed{M}$	$\boxed{M}$	$\boxed{60}$	$\boxed{60}$	$\boxed{64}$ 85	$\boxed{67}$	$\boxed{68}$	$\boxed{84}$	$\boxed{M}$ 48	0
m=3 k=1	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{42}$	$\boxed{40}$ 272	$\boxed{46}$	$\boxed{47}$	$\boxed{M}$ 258	0
k=2	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{63}$	$\boxed{60}$	$\boxed{67}$	$\boxed{67}$	$\boxed{M}$ 133	0
m=4 k=1	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{40}$ 214	$\boxed{40}$ 316	$\boxed{M}$	0
k=2	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{60}$ 38	$\boxed{60}$	$\boxed{M}$ 95	0
D_{jn}	0	0	0	0	0	0	0	0	0	

Figure 4.5

Initial Basic Solution after Phase II

Variable cost \$ 91388 (256 * 40 + 274 * 40 + 2 * 60 + 131 * 68 + 215 * 40 + 275 * 40 + 40 * 44 + 85 * 64 + 272 * 40 + 214 * 40 + 316 * 40 + 38 * 60)

Setup cost \$ 3360 ((396 + 543) + (396 + 543) + 543 + (396 + 543))

Total cost \$ 94748

Period	n=1		n=2		n=3		n=4			C_{km}
	j=1	j=2	j=1	j=2	j=1	j=2	j=1	j=2		
m=1 k=1	40 256	40 274	44	47	48	54	52	61	M	0
k=2	60 2	60	64	77	68 131	84	74	91	M A	0
m=2 k=1	M	M	40 215	40 275	44 40	47	48	54	M	0
k=2	M	M	60	60	64 85	67	68	84	M 48	0
m=3 k=1	M	M	M	M	42	40 272	46	47	M 258	0
k=2	M	M	M	M	63	60	67	67	M 133	0
m=4 k=1	M	M	M	M	M	M	40 214	40 316	M	0
k=2	M	M	M	M	M	M	60 38	60	M 95	0
D_{jn}	0	0	0	0	0	0	0	0	0	

Figure 4.6

Tableau before first iteration of Phase III

As shown in Figure 4.6, Phase III involves applying simplex method to the transportation tableau. This phase chooses non-basic cell A to enter the basis. The entrance of cell A into the basis forces the exit of the cell with 48 units. The tableau after the first iteration in Phase III is shown in Figure 4.7. The final tableau is given in Figure 4.8.

Period	$n=1$		$n=2$		$n=3$		$n=4$			C_{km}
	$j=1$	$j=2$	$j=1$	$j=2$	$j=1$	$j=2$	$j=1$	$j=2$		
$m=1$ $k=1$	$\boxed{40}$ 256	$\boxed{40}$ 274	$\boxed{44}$	$\boxed{47}$	$\boxed{48}$	$\boxed{54}$	$\boxed{52}$	$\boxed{61}$	$\boxed{M}$	0
$k=2$	$\boxed{60}$ 2	$\boxed{60}$	$\boxed{64}$	$\boxed{77}$	$\boxed{68}$ 83	$\boxed{84}$	$\boxed{74}$	$\boxed{91}$	$\boxed{M}$ 48	0
$m=2$ $k=1$	$\boxed{M}$	$\boxed{M}$	$\boxed{40}$ 215	$\boxed{40}$ 275	$\boxed{44}$ 40	$\boxed{47}$	$\boxed{48}$	$\boxed{54}$	$\boxed{M}$	0
$k=2$	$\boxed{M}$	$\boxed{M}$	$\boxed{60}$	$\boxed{60}$	$\boxed{64}$ 133	$\boxed{67}$	$\boxed{68}$	$\boxed{84}$	$\boxed{M}$	0
$m=3$ $k=1$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{42}$	$\boxed{40}$ 272	$\boxed{46}$	$\boxed{47}$	$\boxed{M}$ 258	0
$k=2$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{63}$	$\boxed{60}$	$\boxed{67}$	$\boxed{67}$	$\boxed{M}$ 133	0
$m=4$ $k=1$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{40}$ 214	$\boxed{40}$ 316	$\boxed{M}$	0
$k=2$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{60}$ 38	$\boxed{60}$	$\boxed{M}$ 95	0
D_{jn}	0	0	0	0	0	0	0	0	0	

Figure 4.7

Tableau after first iteration of Phase III

Variable cost of the problem	$ 91196
Setup cost of the problem	$ 3360
Total cost of the problem	$ 94556

Costs changed after the first iteration of Phase III

Variable cost changed	(83 * 68 + 133 * 64) - (131 * 68 + 85* 64) = -$ 192
Setup cost changed	$ 0
Total cost changed	-$ 192

Period	n=1		n=2		n=3		n=4			C_{km}
	j=1	j=2	j=1	j=2	j=1	j=2	j=1	j=2		
m=1 k=1	$\boxed{40}$ 256	$\boxed{40}$ 274	$\boxed{44}$	$\boxed{47}$	$\boxed{48}$	$\boxed{54}$	$\boxed{52}$	$\boxed{61}$	$\boxed{M}$	0
k=2	$\boxed{60}$ 2	$\boxed{60}$	$\boxed{64}$	$\boxed{77}$	$\boxed{68}$	$\boxed{84}$	$\boxed{74}$	$\boxed{91}$	$\boxed{M}$ 131	0
m=2 k=1	$\boxed{M}$	$\boxed{M}$	$\boxed{40}$ 215	$\boxed{40}$ 275	$\boxed{44}$ 36	$\boxed{47}$	$\boxed{48}$	$\boxed{54}$	$\boxed{M}$ 4	0
k=2	$\boxed{M}$	$\boxed{M}$	$\boxed{60}$	$\boxed{60}$	$\boxed{64}$	$\boxed{67}$	$\boxed{68}$	$\boxed{84}$	$\boxed{M}$ 133	0
m=3 k=1	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{42}$ 220	$\boxed{40}$ 272	$\boxed{46}$ 38	$\boxed{47}$	$\boxed{M}$	0
k=2	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{63}$	$\boxed{60}$	$\boxed{67}$	$\boxed{67}$	$\boxed{M}$ 133	0
m=4 k=1	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{40}$ 214	$\boxed{40}$ 316	$\boxed{M}$	0
k=2	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{M}$	$\boxed{60}$	$\boxed{60}$	$\boxed{M}$ 133	0
D_{jn}	0	0	0	0	0	0	0	0	0	

Figure 4.8

Final solution after Phase III

After 9 iterations of Phase III, the optimal solution come out,

Variable cost of the problem	\$ 85056
Setup cost of the problem	\$ 3756
Total cost of the problem	\$ 88812

Total cost difference between the solution of Phase II and Phase III

= \$ 94748 - \$ 88812

= \$ 5936

4.4 Computational Results

We discuss the performance of our heuristic with respect to (1) seasonality, (2) setup costs and (3) tightness of capacity. The details of the design of the experiments are shown in the previous chapter. To analyse the performance of the heuristic, the solution value of our heuristic is compared with the solution of the Gilbert and Madan's heuristic.

Totally randomly generated problems of 27 (3^3) different cases were solved by both our heuristic and the heuristic of Gilbert and Madan. And for each case 30 problems were solved repeatedly to obtain the average performance.

The performance of the heuristic is evaluated by comparing the objective function value of both our heuristic solution and the heuristic solution of Gilbert and Madan. The detailed results of the comparison are shown in Table 4.1. In all 27 cases, our heuristic generated better solutions than the Gilbert and Madan's algorithm. On average, we improved the objective function value by 1.32% better than the Gilbert and Madan's algorithm.

Further, we compare the performance of our algorithm with the Gilbert and Madan's algorithm under different problem characteristics. The results are shown in Tables 4.2 to 4.7. In each comparison, Average Percentage Difference (A.P.D.) is calculated. In Table 4.2, our algorithm performed much better in Medium and High setup costs. For those cases with extreme seasonality, our algorithm performed much better than theirs (see Table 4.3). Table 4.4 shows the effect of tightness of capacity on the A.P.D.. We can see that our algorithm also performed much better in 100% and 120% capacity.

Table 4.5 shows the effect of interaction between setup cost and the seasonality on our heuristic against the Gilbert and Madan algorithm. Our heuristic performed much better than the Gilbert and Madan heuristic. We also made an analysis on the effect of interaction between seasonality and tightness of capacity (see Table 4.6). Our algorithm performed much better than the Gilbert and Madan algorithm, especially in those cases with extreme seasonality and more availability of capacity (120%). Our heuristic also performed much better than the Gilbert and Madan heuristic in the cases which have high setup cost and 120% capacity.

In addition, we also tried to analyse the performance of our new heuristic with respect to the number of products and number of periods. The results are shown in the Table 4.8 and Table 4.9. We found that there were minor effects of the number of products and the number of periods on the difference between our heuristic and the Gilbert and Madan heuristic.

Seasonality	**Setup**	**Capacity**	**Difference ***	**Percentage * Difference (%)**
No	Low	80%	446	0.08
No	Low	100%	2269	0.45
No	Low	120%	3323	0.66
No	Medium	80%	10874	1.76
No	Medium	100%	6876	1.14
No	Medium	120%	8609	1.47
No	High	80%	11553	1.03
No	High	100%	14421	1.53
No	High	120%	18434	2.14
Moderate	Low	80%	825	0.15
Moderate	Low	100%	3055	0.59
Moderate	Low	120%	2687	0.53
Moderate	Medium	80%	15042	2.43
Moderate	Medium	100%	4770	0.79
Moderate	Medium	120%	4959	0.84
Moderate	High	80%	9650	0.84
Moderate	High	100%	10527	1.11
Moderate	High	120%	21587	2.53
Extreme	Low	80%	3515	0.58
Extreme	Low	100%	1007	0.18
Extreme	Low	120%	292	0.06
Extreme	Medium	80%	1141	0.17
Extreme	Medium	100%	4753	0.76
Extreme	Medium	120%	6784	1.13
Extreme	High	80%	10840	1.08
Extreme	High	100%	49366	5.45
Extreme	High	120%	54442	6.21

Difference: Objective Function Value of Gilbert and Madan algorithm OFV_1 minus Objective Function Value of Our algorithm OFV_2

Difference %: $\dfrac{OFV_1 - OFV_2}{OFV_2} * 100\%$

Table 4.1

Average difference and difference percentage of the 27 problem cases between our heuristic and Gilbert and Madan's heuristic

	Setup Cost		
	Low	*Medium*	*High*
A.P.D.	0.36	1.16	2.43

Table 4.2

The effect of setup costs on A.P.D.

	Seasonality		
	None	*Moderate*	*Extreme*
A.P.D.	1.14	1.09	1.73

Table 4.3

The effect of seasonality on A.P.D.

	Capacity		
	80%	*100%*	*120%*
A.P.D.	0.90	1.33	1.73

Table 4.4

The effect of tightness of capacity on A.P.D.

Setup	**Seasonality**		
	None	*Moderate*	*Extreme*
Low	0.40	0.43	0.27
Medium	1.45	1.35	0.69
High	1.57	1.49	4.25

Table 4.5

The effect of seasonality and setup costs on A.P.D.

Capacity	**Seasonality**		
	None	*Moderate*	*Extreme*
80%	0.96	1.14	0.61
100%	1.04	0.83	2.13
120%	1.42	1.30	2.47

Table 4.6

The effect of seasonality and tightness of capacity on A.P.D.

Capacity	**Setup**		
	Low	*Medium*	*High*
80%	0.27	1.45	0.98
100%	0.41	0.89	2.70
120%	0.42	1.15	3.63

Table 4.7

The effect of setup costs and tightness of capacity on A.P.D.

no. of products	3	6	9
A.P.D.	1.32	1.52	1.92

Table 4.8

The effect of number of products on A.P.D.

no. of periods	12	24	36
A.P.D.	1.32	1.24	1.29

Table 4.9

The effect of number of periods on A.P.D.

4.5 Conclusion

In this chapter, we formulated the SLCR problem wihout backordering as a type of the fixed charge transportation problem. The formulation has the advantage of considering overtime capacity. We proposed a solution algorithm exploiting the structure of the fixed charge transportation problem. The algorithm is compared very favorably to the Gilbert and Madan (1991) algorithm for the same problem.

Chapter 5

Conclusion

In this research we studied a class of Single Level Constrained Resource Problem. The problem is formulated as a variation of fixed charge transportation problem. By introducing the new ideas of holding period window size and searching directions, we developed an efficient heuristic on this formulation to solve the SLCR problem with regular time production costs, overtime production costs, inventory holding costs and the fixed setup costs. Our heuristic performed better than the Gilbert and Madan's heuristic especially for those problems with high setup costs, and extreme seasonality.

Besides, we developed a new heuristic for the SLCR problem with backordering which few researchers have worked on in the past. The performance of the heuristic was compared to the Lower Bound we computed. The heuristic solution was satisfactory and had outstanding results for those problems with low setup costs and extreme seasonality.

Our heuristics proposed here performed only on simulated data generated by an experiment. It will be interesting to test our heuristics with real data. Also to make our heuristic truly useful, we have to build a planning system with a user friendly interface.

Another extension to this research, is to incorporate product shelf life. Many products particularly food, beverage, etc. have expiry date. Our heuristics have to be modified if shelf life is a critical issue.

Bibliography

[1] Aras, Omar A. and Swanson, L.G., A Lot Sizing and Sequencing Algorithm for Dynamic Demands upon a Single Facility, *Journal of Operations Management*, 2(3), 177-185 (May 1982)

[2] Bahl, H.C., Column Generation Based Heuristic for Multi-Item Scheduling, *AIIE Transactions*, 15(2), 136-141 (June 1983)

[3] Bahl, H.C. and Ritzman, L.P.,A Cyclical Scheduling Heuristic for Lot Sizing with Capacity Constraints, *International Journal of Production Research*, 22(5), 791-800 (1984)

[4] Balinski, M.L., Fixed Cost Transportation Problem, *Naval Research Logistics Quarterly*, 8(1), 41-54 (1961)

[5] Barany, I., Van Roy, T.J. and Wolsey, L.A., Strong Formulations for Multi-Item Capacitated Lot Sizing, *Management Science*, 30(10), 1255-1261 (October 1984)

[6] Barr, R.S., Glover, F. and Klingman, D., A New Optimization Method for Large Scale Fixed Charge Transportation Problems, *Operations Research*, 29(3), 448-463 (May-June 1981)

[7] Berry, W.L., Lot Sizing Procedures for Requirements Planning Systems: A Framework for Analysis, *Production and Inventory Management*, 13(2), 19-34 (1972)

[8] Billington, P.J., McClain, J.O., and Thomas, L.J., Mathematical Programming Approaches to Capacity Constrained MRP Systems: Review, Formulation, and Problem Reduction, *Management Science*, 29(10), 1126-1141 (October 1983)

[9] Bitran, G.R. and Yanassee, H.H., Computational Complexity of the Capacitated Lot Sizing Problem, *Management Science*, 28(10), 1174-1186 (October 1982)

[10] Blackburn, J.D. and Millen, R.A., Heuristic Lot-Sizing Performance in a Rolling Schedule Environment, *Decision Science*, 11(4), (October 1980)

[11] Blackburn, J.D. and Millen, R.A., Improved Heuristics for Multi-Stage Requirements Planning Systems, *Management Science*, 28(1), 44-56 (January 1982)

[12] Bradley, G.H., Brown, G.G., and Graves, G.W., Design and Implementation of Large Scale Primal Transshipment Algorithms, *Management Science*, 24(1), 1-33 (September 1977)

[13] Bradley, G.H., Brown, G.G., and Graves, G.W., GNET, A Primal Capacitated Network Program, Copyright 1975, 1977

[14] Buffa, E.S. and Miller, J.G., *Production-Inventory Systems: Planning and Control*, Richard D. Irwin, Homewood, Illinois (1979)

[15] Cabot, A.V. and Erenguc, S.S., Some Branch-and-Bound Frocedures for Fixed-Cost Transportation Problems, *Naval Research Logistics Quarterly*, 31, 145-154 (1984)

[16] Cooper, L. and Drebes, C., An Approximate Solution for the Fixed Charge Problem, *Naval Research Logistics Quarterly*, 14(1), 101-113 (1967)

[17] Dantzig, G.B. and Wolfe, P., A Decomposition Algorithm for Linear Programs, *Econometrica*, 29, 767-778 (1961)

[18] Denzler, D.R., An Approximate Algorithm for the Fixed Charge Problem, *Naval Research Logistics Quarterly*, 16(3), 411-416 (1969)

[19] Diaby, M, Successive Linear Approximation Procedure for Generalized Fixed-Charge Transportation Problems, *Journal of the Operational Research Society*, 42(11), 991-1002 (1991)

[20] Dixon, P.S. and Silver, E.A., A Heuristic Solution Procedure for the Multi-Item, Single-Level, Limited Capacity, Lot-Sizing Problem, *Journal of Operations Management*, 2(1), 23-29 (October 1981)

[21] Dogramaci, A, Panayiotopoulos, J.E., and Adam, N.R., The Dynamic Lot Sizing Problem for Multiple Items Under Limited Capacity, *AIIE Transactions*, 13(4), 294-326 (December 1981)

[22] Driebeek, N.J., An Algorithm for the Solution of Mixed Integer Programming Problems, *Management Science*, 9(2), 229-258 (January 1963)

[23] Dzielinski, B.P., Baker, C.T., and Manne, A.S., Simulation Tests of Lot Size Programming, *Management Science*, 9(2), 229-258 (January 1963)

[24] Dzielinski, B.P. and Gomory, C.T., Optimal Programming of Lot Sizes, Inventory and Labor Allocations, *Management Science*, 2(9), 874-890 (July 1965)

[25] Eisenhut, P.S., A Dynamic Lot-Sizing Algorithm with Capacity Constraints, *AIIE Transactions*, 7(2), 170-176 (June 1975)

[26] Elmaghraby, S.E., The Economic Lot Scheduling Problem (ELSP): Review and Extension, *Management Science*, 24(6), 587-598 (1978)

[27] Eppen, G. and Martin, R., Multi-item Capacitated Lot Sizing using Variable Redefinition, *Operations Research*, 35, 832-848 (1987)

[28] Fisk, J., An Initial Bounding Procedure for use with 0-1 Single Knapsack Algorithms, *Operations Research*, 14, 88-98 (1977)

[29] Fisk, J. and McKeown, P., The Pure Fixed Charge Transportation Problem, *Naval Research Logistics Quarterly*, 26, 631-641 (1979)

[30] Florian, M. and Klein, M., Deterministic Production Planning with Concave Costs and Capacity Constraints, *Management Science*, 18(1), 12-20 (September 1971)

[31] Florian, M., Lenstra, J.K. and Rinnooy Kan, A.H.G., Deterministic Production Planning: Algorithms and Complexity, *Management Science*, 26(7), 12-20 (July 1980)

[32] Gabbay, H., Multi-Stage Production Planning, *Management Science*, 25(11), 1138-1148 (November 1979)

[33] Garey, M.R. and Johnson, D.S., *Computations and Tractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, California, (1979)

[34] Garfinkel, R.S. and Nemhauser, G.L., *Integer Programming*, John Wiley and Sons, New York (1972)

[35] Gary, P., Exact Solution of the Fixed-Charge Transportation Problem, *Operations Research*, 19(6), 1629-1538 (August 1968)

[36] Gilbert, K.C. and Madan, M.S., A Heuristic for a Class of Production Planning and Scheduling Problems, *IIE Transactions*, 23(3) 282-289 (1991)

[37] Graves, S.C., Multi-Stage Lot Sizing: An Iterative Procedure, in Schwartz, L.B., (Ed.), *Multi-Level Production/Inventory Control System: Theory and Practice*, North-Holland, 95-109 (1981)

[38] Graves, S.C., Using Lagrangean Techniques to Solve Hierarchial Production Planning Problems, *Management Science*, 28(3), 260-275 (March 1982)

[39] Haehling von Lanzenhauer, C., Production and Employment Scheduling in Multi-Stage Production Systems, *Naval Research Logistics Quarterly*, 17(2), 193-198 (June 1970)

[40] Haessler, R.W., An Improved Extended Basic Period Procedure for Solving the Economic Lot Scheduling Problem, *AIIE Transactions*, 2, 336-340 (1979)

[41] Hindi, K.S., Computational Efficient Solution of the Multi-Item, Capacitated Lot-Sizing Problem, *Computers and Industrial Engineering*, 28(4), 709-719 (1995)

[42] Hirsch, W.M. and Dantzig, G.B., The Fixed Charge Problem, *Naval Research Logistics Quarterly*, 15(3), 413-424 (1968)

[43] Karni, R., Maximum Part-Period Gain: A Lot-Sizing Procedure for Unconstrained and Constrained Requirement Planning Systems, *Production and Inventory Management*, 22(2), (1981)

[44] Karni, R. and Roll, Y., A Heuristic Algorithm for the Multi-Item Lot-Sizing Problem with Capacity Constraints, *AIIE Transactions*, 14(4), 249-256 (1982)

[45] Kennington, J., The Fixed-Charge Transportation Problem: A Computational Study with a Branch-and-Bound Code, *AIIE Transactions*, 8(2), 241-247 (1976)

[46] Kennington, J. and Unger, E., A New Branch-and-Bound Algorithm for the Fixed-Charges Transportation Problem, *Management Science*, 22(10), 1116-1126 (1976)

[47] Huhn, H.W. and Baumol, W.J., An Approximate Algorithm for the Fixed-Charges Transportation Problem, *Naval Research Logistics Quarterly*, 9(1), 1-15 (July 1961)

[48] Lambrecht, M.R. and Vander Eecken, J., A Facilities in Series Capacity Constrained Dynamic Lot-Sizing Model, *European Journal of Operational Research*, 2(1), 42-49 (1978a)

[49] Lambrecht, M.R. and Vander Eecken, J., Capacity Constrained Single-Facility Dynamic Lot-Sizing Model, *European Journal of Operational Research*, 2(2), 132-136 (1978b)

[50] Lambrecht, M.R. and Vanderveken, H., Heuristic Procedures for the Single Operation, Multi-Item Loading Problem, *AIIE Transactions*, 11(4), 319-326 (December 1979)

[51] Lasdon, L.S. and Terjung, R.C., An Efficient Algorithm for Multi-Item Scheduling, *Operations Research*, 19, 892-914 (1967)

[52] Manne, A.S., Programming of Economic Lot Sizes, *Management Science*, 4(2), 115-135 (January 1958)

[53] McKeown, P.G., A Branch-and-Bound Algorithm for Solving Fixed Charge Problems, *Naval Research Logistics Quarterly*, 28(4), 607-617 (December 1981)

[54] McKeown, P.G. and Ragsdale, C.T., A Computational Study of using Preprocessing and Stronger Formulation to solve large General Fixed-Charge Problems, *Computer Operations Research*, 17(1), 9-16 (1990)

[55] Meal, H.C. and Silver, E.A., A Simple Modification of the EOQ for the Case of a Varying Demand Rate, *Production and Inventory Management*, (Fourth Quarter 1969)

[56] Millar, H. and Yang, M., Lagrangain Heuristic for the Capacitated Multi-Item Lot-Sizing Problem with Backordering, *International Journal of Production Economics*, 34, 1-15 (1994)

[57] Murty, K.G., Solving the Fixed Charge Problem by Ranking Extreme Points, *Operations Research*, 16(2), 268-279 (March-April 1968)

[58] Newson, E.F.P., Multi-Item Lot Size Scheduling by Heuristic-Part 1: with Fixec Resources and Part 2: with Variable Resources, *Management Science*, 21(10), 1186-1203 (June 1975)

[59] Palekar, U.S., Karwan, M.H. and Ziouts, S., A Branch and Bound Method for the Fixed-Charge Transportation Problem, *Management Science*, 26(9), 1092-2002 (1990)

[60] Pochet, Y. and Wolsey, L., Lot Size Models with Backlogging: Strong Reformulations and Cutting Planes, *Mathematical Programming*, 40, 317-335 (1988)

[61] Posner, M.E. and Szwarc, W., A Transportation Type Aggregate Production Model with Backordering, *Management Science*, 29(2), 188-199 (February 1983)

[62] Reinfeld, N.V. and Vogel, W.R., *Mathematical Programming*, Prentice-Hall, Englewood Cliffs, N.J., 1958

[63] Ross, G.J. and Szwarc, W., A Transportation Type Aggregate Production Model with Backordering, *Management Science*, 29(2), 188-199 (February 1983)

[64] Schweitzer, P.J. and Silver, E.A., Mathematical Pitfalls in the One Machine Multiproduct Economic Lot Scheduling Problem, *Operations Research*, 31(2), 401-405 (March-April 1983)

[65] Silver, E.A. and Meal, H.C., A Heuristic for Selecting Lot Size Quantities for the Case of a Deterministic Time Varying Rate and Discrete Opportunities for Replenishment, *Production and Inventory Management*, 14(2), 64-74 (1973)

[66] Steinberg, D.I., The Fixed Charge Problem, *Naval Research Logistics Quarterly*, 17(2), 217-235 (1970)

[67] Steinberg, E. and Napier, H.A., Optimal Multi-Level Lot Sizing for Requirement Planning Systems, *Management Science*, 26(12), 1258-1271 (1980)

[68] Thizy, J.M. and Van Wassenhove, L.N., A Subgradient Algorithm for the Multi-Item Capacitated Lot Sizing Problem, *AIIE Transactions*, 17(4), 308-313 (December 1985)

[69] Tomlin, J.A., An Improved Branch-and-Bound Method for Integer Programming, *Operations Research*, 19(4), 1070-1075 (November-December 1971)

[70] Trotter, L.E. and Shetty, C.M., An Algorithm for the Bounded variable Integer Programming Problem, *Journal of the Association for Computing Machinery*, 21, 505-513 (1974)

[71] Wagner, H.M. and Whitin, T.M., Dynamic Version of the Economic Lot Size Model, *Management Science*, 5(1), 89-96 (Octocber 1958)

[72] Walker, W.E., A Heuristic Adjacent Extreme Point Algorithm for the Fixed Charge Problem, *Management Science*, 22(5), 587-596 (January 1976)

[73] Wright, D. Lanzenaneuvon and Haehling, C., COAL: A New Heuristic Approach for solving the Fixed-Charge Problem - Computational Results, *European Journal of Operational Research*, 52(2), 235-247 (1991)

[74] Zahorik, A., Thomas, L.J. and Trigeiro, W.W., Network Programming Models for Production Scheduling in Multi-Stage Multi-Item Capacitated Systems, *Management Science*, 30, 308-325 (1984)

[75] Zangwill, W.I., A Deterministic Multi-Product, MultiFacility, Production and Inventory Model, *Operations Research*, 14(3), 486-507 (May-June 1966)

[76] Zangwill, W.I., A Backlogging Model and a Multi-Echelon Model of a Dynamic Economic Lot-Size Production System - A Network Approach, *Management Science*, 15(9), 506-597 (May 1969)

[77] Zangwill, W.I., Minimum Concave Cost Flows in Certain Networks, *Management Science*, 14(7), 429-450 (March 1968)